UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Classificació d'àudio DCASE Challenge

Treball Final de Grau

Grau en Enginyeria de Sistemes Audiovisuals

Autor: Alex Bru Buxó
Director: Ignasi Esquerra Llucià

Universitat Politècnica de Catalunya (UPC)
Juny de 2019

Resum

En els últims anys, l'ús de les xarxes neuronals ha esdevingut el pilar fonamental en el desenvolupament de sistemes de classificació d'escenes acústiques (ASC). Aquesta nova base ha impulsat l'expansió dels límits tecnològics d'aquest àmbit.

En aquest projecte, es realitza l'estudi i la implementació d'un sistema que permet classificar 9 activitats domèstiques (prèviament definides). L'estructura està formada per dos components: una extracció de característiques de coeficients MFCC, els quals permeten realitzar l'anàlisi sobre les característiques freqüencials de les classes i, a continuació, una xarxa neuronal formada per capes CNN i Fully-Connected. L'entrenament del sistema es realitza a partir de l'algoritme d'Adam, una combinació dels algoritmes d'AdaGrad i RMSProp, i una estructura de 4-Fold Cross-Validation, que permet avaluar els resultats del sistema. El resultat final del sistema és d'un 84,63% d'encert per a la mitjana dels 4 *folds*.

A partir de l'estudi realitzat s'han implementat modificacions en l'extracció de característiques, les quals mostren les propietats bàsiques de les escenes a classificar, a la vegada que es milloren els resultats de 6 de les 9 classes estipulades.

Abstract

In recent years, the use of neural networks has become the fundamental pillar in the development of acoustic scene classification systems (ASC). This new ground has led to the expansion of technological boundaries in this area.

In this project, the study and implementation of a system is used to classify 9 domestic activities (previously defined). The structure consists of two components: an extraction of MFCC characteristics, which allows the analysis of freqüential characteristics of the classes, and then a neural network, formed by CNN and Fully-Connected layers. System training is based on the Adam algorithm, a combination of the AdaGrad and RMSProp algorithms, and a 4-Fold Cross-Validation structure, which allows for the evaluation of the system. The final result of the system is 84.63% for the average of 4 folds.

From the study, modifications are implemented to the extraction of features, which show the basic properties of the scenes to be classified, while improving the results of 6 of the 9 classes stipulated.

Índex

Llistat d'Il·lustracions

Llistat de Taules

Llistat d'Abreviacions

ANN	Artificial Neural Network	Xarxa Neuronal Artificial
API	Application Programming Interface	Interfàs de Programació d'Aplicació
ASC	Acoustic Scene Classification	Classificació d'Escenes Acústiques
CNN	Convolutional Neural Network	Xarxa Neuronal Convolucional
DCASE	Detection and Classification of Acoustic Scenes and Events	Detecció i Classificació d'Escenes i Esdeveniments Acústics
DCT	Discrete Cosine Transform	Transformada Discreta de Cosinus
DNN	Deep Neural Network	Xarxa Neuronal Profunda
FFT	Fast Fourier Transform	Transformada Ràpida de Fourier
HAS	Human Auditory System	Sistema Audiu Humà
MFCC	Mel Frequency Cepstral Coefficients	Coeficients Cepstrals en l'Escala Mel
MLP	Multy-Layer Perceptron	Perceptró Multicapa
RNN	Recurrent Neural Network	Xarxa Neuronal Recurrent
SGD	Stochastic Gradient Descent	Gradient Descendent Estocàstic
SVM	Support-Vector Machine	Màquines de Vectors de Suport
TSC	Signal Theory and Communications Department	Departament de Teoria del Senyal

Capítol 1

Introducció

Recentment, s'ha elevat un gran interès en el monitoratge i la millora del benestar i la qualitat de vida de les persones dins de la llar a través de diferents sensors, com el cas del micròfon. El monitoratge dins de casa permet donar suport, seguretat i confort a les persones (p. ex. pacients amb malalties cròniques, gent gran, etc.). Per tal d'aconseguir una correcta funcionalitat del sistema, és necessari el precís reconeixement de la situació. Per motius com aquests, la detecció i classificació d'escenes i esdeveniments acústics ha guanyat un gran interès en la comunitat científica, i ha permès la publicació de diverses bases de dades per a un gran nombre d'aplicacions. Per altra banda, també s'ha popularitzat el desenvolupament d'assistents vocals com Google Home, Apple HomePod i Amazon Echo, que fins al moment, només tenen el punt de mira en el reconeixement de la parla, però podrien ser ampliats a la identificació d'activitats domèstiques. D'aquesta manera, sorgeixen comunitats com DCASE, que permeten expandir els límits d'aquestes noves tecnologies.

En les últimes dècades, components com els *wearables* han anat reduint la seva mida, permetent l'obtenció d'informació espacial. A la vegada, la majoria de models de classificació d'escenes acústiques (ASC) (Virtanen, Plumbley and Ellis, 2017) estan desenvolupats sota les mateixes característiques: enregistrament d'un sol canal en una localització fixa i única. La tasca 5 del repte de DCASE 2018 (Dekkers *et al.*, 2018), en canvi, procura investigar l'impacte d'enregistraments multicanal en el reconeixement d'escenes acústiques domèstiques, a la vegada que incrementa la resolució espacial, a partir de la base de dades SINS (Dekkers *et al.*, 2017), que proporciona sensors distribuïts per tota la llar. Cal tenir en compte que l'ASC intenta resoldre un problema complex a causa de la gran varietat de sons i esdeveniments que tenen lloc en una escena acústica, mentre que només una petita part d'aquests dóna informació rellevant sobre l'escena. Aquest repte, però, proporciona una base de dades d'estudi sense superposició d'activitats, organitzada i etiquetada. Conjuntament amb els àudios, els organitzadors del repte comparteixen un sistema classificador basat en coeficients MFCC i una estructura neuronal de Deep Learning, que resulta ser molt eficient comparat amb l'estat de l'art actual.

1.1 Objectiu

L'objectiu d'aquest projecte consisteix en la implementació, l'estudi i la millora o modificació del sistema proporcionat per la tasca 5 de DCASE Challenge 2018, que té com a finalitat la classificació d'escenes acústiques domèstiques, com per exemple, cuinar o mirar la televisió, i que han sigut enregistrades per diversos vectors de micròfons en diverses localitzacions de la llar.

El sistema Baseline, proporcionat pels organitzadors, està dividit en dues grans estructures: primerament, una extracció de característiques MFCC (4.2.1), les quals s'ofereixen com a entrada a la segona estructura, una xarxa neuronal artificial, formada per conjunts de CNNs i capes Fully-

Connected, que és entrenada a través d'algoritmes de Deep Learning (4.2.3). La classificació de les escenes acústiques està limitada a 9 classes diferents (2.2).

Per a la implementació i l'estudi del sistema Baseline es dividirà el projecte en dues grans etapes. Una primera fase realitzarà un estudi de la base de dades i les característiques bàsiques dels àudios. En la segona etapa es posarà en marxa el sistema Baseline, i es realitzaran les modificacions pertinents per a la correcta execució. A partir de l'execució del sistema, serà possible l'anàlisi dels resultats obtinguts (Capítol 5).

Aquest projecte està dirigit pel Departament de Teoria del Senyal (TSC) de l'ESEIAAT, de la Universitat Politècnica de Catalunya (UPC), i desenvolupat en els servidors de VEU, del Grup de Tractament de la Parla (TALP).

Capítol 2

DCASE Challenge

2.1 Què és el repte DCASE?

El repte de DCASE consisteix en el desenvolupament de mètodes d'anàlisi computacionals d'escenes i esdeveniments acústics a través de la comparació de diferents propostes, utilitzant bases de dades públiques i disponibles, que permeten mesures del rendiment comparables. Es proposen de forma anual diverses tasques per tal de millorar la classificació i detecció d'aspectes de l'acústica ambiental. A través d'aquesta organització es desenvolupen sistemes pioners, a la vegada que es solidifica el rendiment per a futures referències (DCASE Community, 2019).

2.1.1 Tasques del repte

El repte DCASE Challenge de 2018 es va organitzar entre el 30 de març i el 31 de juliol de 2018. Va comptar amb les següents tasques:

- Tasca 1: Classificació d'escenes acústiques del carrer.
- Tasca 2: Classificació d'àudios generalistes amb contingut de Freesound (Freesound, 2019) i amb etiquetes d'AudioSet (AudioSet, 2019).
- Tasca 3: Detecció d'àudio d'ocells.
- Tasca 4: Detecció d'esdeveniments en entorns domèstics semisupervisats amb dades superposades i escassament etiquetades.
- Tasca 5: Monitoratge d'activitats domèstiques basat en acústica multicanal (2.1.2).

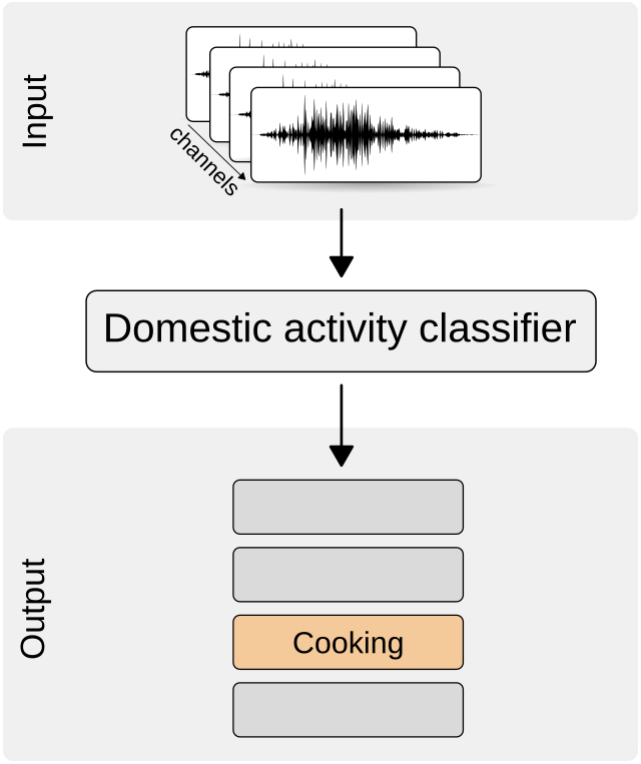
La principal diferència de la Tasca 5 (2.1.2), respecte a la Tasca 1, es mostra en la classe d'elements a classificar, així com la possibilitat d'utilitzar àudio multicanal i independència de la localització.

2.1.2 Definició de la Tasca 5

Actualment, la major part dels models acústics d'estudi que permeten la identificació d'activitats domèstiques, es basen en gravacions d'un sol canal i en una sola localització. En aquesta tasca es planteja el següent repte d'estudi: investigar els beneficis i l'efectivitat que comporta l'ampliació del nombre de canals acústics i diversificació de la posició durant l'enregistrament, en la detecció d'activitats domèstiques.

Per a la realització d'aquesta tasca, s'ofereix com a repte, la classificació de segments d'àudio multicanal (i.e. es proporcionen les dades segmentades), adquirits a través d'un vector de micròfons, en una de les classes predefinides (*Il·lustració 2.1*). Les classes a classificar són

activitats diàries realitzades en un entorn domèstic; com per exemple, cuinar, mirar la televisió o treballar.

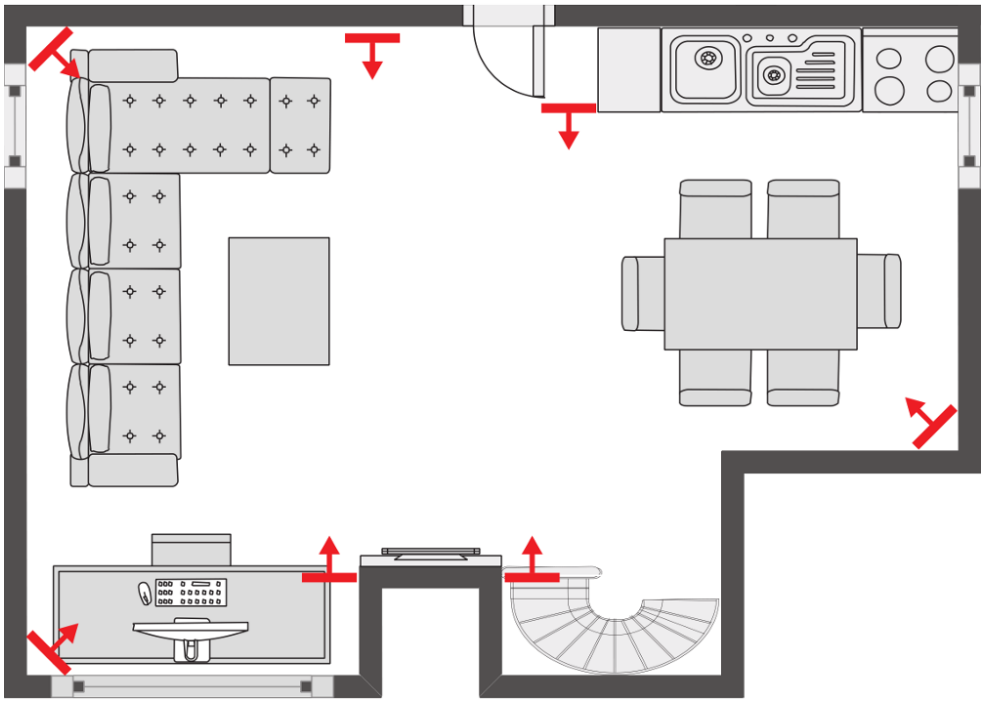


Il·lustració 2.1: Estructura general de la tasca (Dekkers *et al.*, 2018).

En aquest repte es redueix la complexitat de l'estudi al reduir a una el nombre de persones sobre les quals s'analitzen les activitats, característica que permet remetre la superposició d'activitats, ja que la base de dades està tractada per no tenir dades d'activitats superposades. Aquestes condicions permeten focalitzar l'objectiu de la tasca, i permeten integrar les característiques espacials de l'enregistrament com a entrada del sistema classificador. Considerar una posició absoluta en les fonts d'entrada per a la detecció del model condiciona negativament en la generalització correcta dels casos on la posició del vector és alterada. Per aquest motiu, l'objectiu de la tasca se centra en sistemes capaços d'explotar les característiques espacials independents de la localització dels sensors, utilitzant àudio multicanal.

2.2 Base de dades

La base de dades utilitzada deriva de la Base de Dades SINS (Dekkers *et al.*, 2017), la qual està formada per l'enregistrament d'àudio continu d'una persona de vacances, durant el període d'una setmana. Va ser enregistrada pel grup de recerca ADVISE (AdvISe lab – Ku Leuven, 2019) a partir d'una xarxa de 13 vectors (també anomenats nodes) de micròfons distribuïts per tota la casa (*Il·lustració 2.2*). Cada vector està format per 4 micròfons disposats de manera lineal. Per realitzar el projecte DCASE s'utilitzen únicament 7 vectors de micròfons, repartits entre el menjador i la cuina. A continuació podem observar la disposició i orientació dels vectors en el pla de la casa, on resideixen els 7 vectors utilitzats.



Il·lustració 2.2: Plànol 2D del menjador i cuina de l'habitatge amb els nodes, de (Dekkers *et al.*, 2018).

Per tal d'obtenir una base de dades el més fiable possible, no s'hi han definit escenaris predefinits ni restriccions sobre les activitats realitzades. Tot i així el nombre d'activitats etiquetades es restringit a 9.

2.2.1 Contingut

En la base de dades, les gravacions contínues han sigut prèviament dividides en segments d'àudio de 10 segons. Els segments que contenien més d'una activitat han sigut descartats (p. ex. transicions entre dues activitats). Per tant, cada segment només representa una sola activitat. També s'ha reduït el nombre de mostres dels segments en les classes més grans, per tal de poder utilitzar la base de dades amb més flexibilitat. Obtenim, finalment, fitxers d'àudio en format 'wav' de 4 canals, on cada canal pertany a un micròfon d'un mateix node o vector. A continuació podem observar una taula amb les activitats diàries a avaluar (*Taula 2.1*), realitzades en l'àrea del menjador i la cuina, així com el nombre total de segments de 10 segons i de sessions pertanyents a cada tipus d'activitat.

Les diferents activitats/classes a avaluar seran anomenades en anglès tal com apareixen en els fitxers de la base de dades: 'Absence', 'Cooking', 'Dishwashing', 'Eating', 'Other', 'Social activity', 'Vacuum cleaning', 'Watching TV' i 'Working'.

Activitat	Núm. Segments 10 seg.	Núm. sessions
Absència (ningú present a l'habitació)	18860	42
Cuinar	5124	13
Rentar plats	1424	10
Menjar	2308	13
Altres (present, però no realitzant una activitat rellevant)	2060	118
Activitat social (visita, trucada)	4944	21
Neteja amb aspiradora	972	9
Mirar televisor	18648	9
Treballar (teclejar, clicar el ratolí)	18644	33
Total	**72984**	**268**

Taula 2.1: Nombre de segments de 10 segons i nombre de sessió per activitat, de (Dekkers *et al.*, 2018).

Per sobre de tot, la base de dades reflecteix el desequilibri sobre les activitats realitzades en el dia a dia, ja que es presenta de forma no balancejada (*Taula 2.1*). Per als casos de "Mirar televisor" i "Absència" hi trobem un factor d'entre 10 i 30 vegades més grans en termes de duració, respecte a les activitats més curtes, com "Neteja amb aspiradora" i "Altres".

La base de dades es divideix en dues parts: desenvolupament i validació. La part de desenvolupament conté les dades de 4 dels 7 vectors de la base de dades, mentre que la part de validació conté dades de tots els 7 nodes, incloent-hi també els 3 nodes no presents en el desenvolupament. Per a aquest projecte únicament s'utilitzaran les dades de desenvolupament, ja que no participem en el repte, i per tant, no disposem de les metadades per a la validació.

La base de dades de desenvolupament conté aproximadament 200 hores d'informació de 4 nodes diferents, així com les metadades necessàries per identificar cada segment. La partició de les dades entre desenvolupament i validació s'ha fet de forma aleatòria, però sí s'han mantingut junts els fitxers d'una mateixa sessió contínua d'una activitat en particular (p. ex. tota una sessió de cuina). Les dades enregistrades per cada node contenen informació del mateix període de temps. És a dir, les activitats realitzades són vistes des dels diferents nodes al mateix temps. A causa de la reducció de mostres de les classes més grans, no es dóna mai una superposició d'una sola activitat en tots els nodes, de qualsevol activitat.

2.2.2 Procediment d'enregistrament i anotació

Una taula de control amb 4 micròfons formen la configuració de cada un dels diferents nodes. La taula de control conté un microcontrolador EFM32 ARM cortex M4 de Silicon Labs (EFM32WG980), que permet mostrejar l'àudio analògic. El vector de micròfons està compost per 4 Sonion N8AC03 MEMS de baixa potència (±17µW), amb una distància entre els micròfons de 5 cm. El mostreig de les dades es realitza seqüencialment a una velocitat de 16 kHz i amb 12 bits. El procediment d'anotació es va realitzar en dues fases. Primerament, durant l'enregistrament, una aplicació de telèfon permetia a la persona que estava sent enregistrada,

anotar les activitats que estaven sent realitzades en aquell moment d'un llistat fixat d'activitats. A continuació, es van definir les marques de temps d'inici i final de cada activitat, que defineixen les transicions. Durant el període de la setmana, no només es va enregistrar a les persones que vivien a la casa, sinó també múltiples visitants, i inclús diverses trucades de telèfon. Tota la informació de compartició i el processament posterior de la base de dades incorpora aspectes relacionats amb la privacitat dels ocupants.

2.2.3 Estructura

Al descarregar la base de dades obtenim els següents fitxers:

- **EULA.pdf** Acord de llicència d'usuari final.
- **meta.txt** Meta data, format 'tsv'.
- **README.md** Descripció de la base de dades (markdown).
- **README.html** Descripció de la base de dades (HTML).
- **Audio/** 72984 segments d'àudio, 16-bit 16kHz.
  - DevNode1_ex1_1.wav
  - DevNode2_ex1_2.wav
  - ...
- **evaluation_setup/** Configuració 4 Fold Cross-Validation.
  - fold1_train.txt Llistat de fitxers d'entrenament, format 'tsv'.
  - fold1_test.txt Llistat de fitxers de test, format 'tsv'.
  - fold1_evaluate.txt Llistat de fitxers d'avaluació, format 'tsv'.
  - ...

Tant l'estructura del fitxer 'meta.txt', com els fitxers dins de la carpeta '/evaluation_setup' tenen la següent estructura: [fitxer àudio (str)][tab][etiqueta (str)][tab][sessió (str)]. Aquesta estructura permet emmagatzemar el nom, així com afegir opcionalment la informació verificada. En la carpeta '/evaluation_setup' hi trobem les dades necessàries per realitzar el 4-Fold Cross-Validation, amb el següent format depenent de la funcionalitat:

- **Entrenament**: [fitxer àudio (str)][tab][etiqueta (str)][tab][sessió (str)].
- **Test**: [fitxer àudio (str)]
- **Avaluació**: [fitxer àudio (str)][tab][etiqueta (str)].

L'estructura per a l'avaluació està dividida en 4 carpetes o *folds*, que distribueixen els fitxers d'àudio disponibles. Els segments d'una mateixa sessió es mantenen junts dins de la mateixa carpeta per evitar la relació entre els *folds*. Per cada un dels *folds* obtenim un conjunt d'entrenament, de test i d'avaluació.

Sota la carpeta '/audio' trobem els diferents fitxers d'àudio amb el següent format de nom: DevNode{NodeID}_ex{sessionID}_{segmentID}.wav. On 'NodeID' (que pren valors d'entre 1 i 4) permet identificar a quin node pertany el segment. No tenim informació de la posició del node. Per altra banda, 'sessionID' indica una sessió completa d'una activitat determinada. I finalment, 'segmentID' enumera els segments dins d'una sessió, ja que una mateixa sessió pot tenir diversos segments. Cal tenir en compte que el 'segmentID' no es comparteix entre els diversos nodes (p. ex. DevNode1_ex1_1 no està necessàriament enregistrat en el mateix període que DevNode2_ex1_1, però per descomptat comparteixen la mateixa sessió.

Capítol 3

Marc teòric

Tenint en compte que l'objectiu de l'estudi del projecte centra gran part del punt de mira en l'arquitectura neuronal del classificador, serà necessari el plantejament d'alguns paràmetres que serveixin com a fonaments conceptuals sobre els quals es basa la lectura del projecte. Inicialment, es definiran els conceptes bàsics sobre el Deep Learning i les Xarxes Neuronals Artificials (3.1, 3.2), fent èmfasi en els conceptes utilitzats durant el projecte, de la mateixa manera que es defineixen en (Schmidhuber, 2014). Finalment, s'exposarà l'objectiu d'estudi i de millora de l'ASC (3.3), a partir dels termes exposats anteriorment.

El Deep Learning (en català, Aprenentatge Profund) forma part d'una àmplia família de mètodes dins del Machine Learning, o Aprenentatge Automàtic. Està basat en les Xarxes Neuronals Artificials (3.1), i compost per un conjunt d'eines i tècniques que, a través de grans bases de dades d'exemple i de gran potència computacional, són capaces d'identificar automàticament patrons complexos de dades. L'aprenentatge pot ser supervisat, semisupervisat o no supervisat (Schmidhuber, 2014). Diverses arquitectures han permès un gran avenç en camps com el de visió per computador i el reconeixement de la parla i d'escenes acústiques.

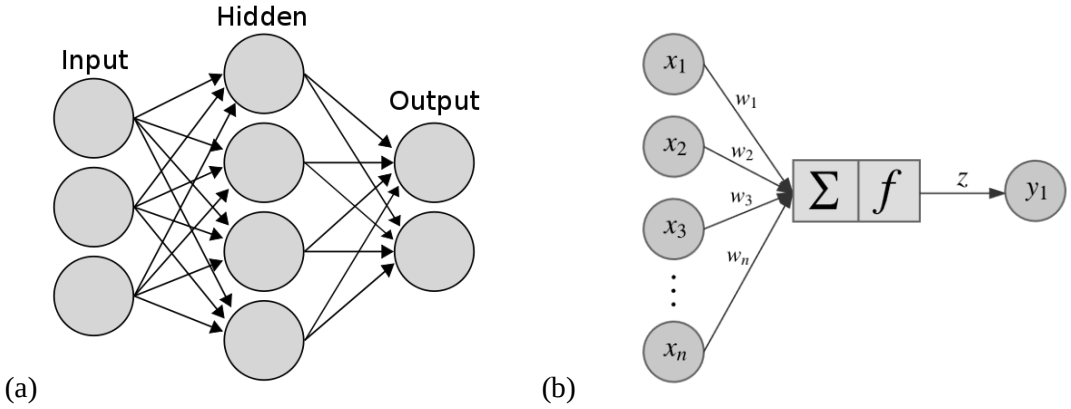
Les tècniques convencionals de Machine Learning han estat sempre limitades en el processament de dades naturals, és a dir, en l'estat més primitiu. Aquest fet imposa a l'extracció de característiques com a un element imprescindible, sensible i de gran pes en la serialització de les dades primitives, que permet extreure representacions adequades per a la seva classificació. A partir de grans models i grans bases de dades etiquetades, el Deep Learning permet mapejar un vector d'entrada a un de sortida (tasca que resulta fàcil per a una persona).

L'aprenentatge profund és capaç de generar funcions de gran complexitat per a la representació aproximada de la realitat, a partir de la composició de mòduls no lineals, generats a partir de l'increment de capes i unitats. Cada transformació permet augmentar el nivell d'abstracció de les dades d'entrada. Les funcions, per tant, són generades a partir de les pròpies dades.

3.1 Xarxa Neuronal Artificial (ANN)

En Deep Learning, l'arquitectura bàsica s'anomena Xarxa Neuronal Artifical (en anglès, Artificial Neural Network) o ANN, inspirada en la biologia de les neurones. Les ANNs estan definides per un conjunt de nodes, també anomenats neurones, en els quals hi té lloc una operació. Aquests nodes formen una estructura totalment connectada i composta per vàries capes o *layers* (*Il·lustració 3.1*). Cada node de cada capa està connectat a cada una de les neurones de la següent capa, però els nodes d'una mateixa capa són totalment independents i no comparteixen cap connexió. Aquesta estructura permet aprofundir la xarxa a partir de l'inserció de les anomenades capes intermèdies o *hidden layers*, que prenen aquest nom, ja que els valors no són observats en l'entrenament. La capa inicial es defineix com la capa d'entrada o *input layer*, i la capa final com a la capa de sortida o *output layer*.

Definim el Perceptró, o Perceptró Monocapa, com a un algoritme capaç de realitzar l'aprenentatge d'un classificador binari. L'acumulació de múltiples d'aquestes capes s'anomena Perceptró Multicapa (Multy-Layer Perceptron), o MLP, i és capaç de resoldre problemes que no són linealment separables (*Virtanen*, Plumbley and Ellis, 2017).



Il·lustració 3.1: (a): Arquitectura d'una Xarxa Neuronal Artificial (ANN) (*File:Artificial neural network.svg*, 2011). (b): Esquema d'un Perceptró Monocapa de n entrades (Dertat, 2017).

A continuació es definirà el procés de les dades a través d'una ANN, prenent com a exemple els càlculs sobre un Perceptró Monocapa, i recorrent els passos necessaris: funcions d'activació (3.2.1) i optimització (3.2.2).

Un node z d'una capa intermèdia o final es defineix a partir de la injecció d'un vector d'entrada $x = \{x_0,\ x_1,\ x_2,\ \dots,\ x_n\}$, que a continuació és ponderat per un conjunt de pesos $w = \{w_1,\ w_2,\ w_3,\ \dots,\ w_n\}$, de tal forma que cada relació d'entrada amb un node té un pes associat. Finalment, s'afegeix la suma del *bias*, que també es correspon amb una entrada per a cada node. Aquest terme permet orientar el resultat de la funció d'activació, així com ajudar al sistema d'entrenament quan les característiques d'entrada són 0.

$$z = f(b + x \cdot w) = f\left(b + \sum_{i=1}^{n} x_i w_i\right) \qquad (3.1)$$
$$x \in d_{1\times n}, x \in d_{n\times 1}, z \in d_{1\times 1}, b \in d_{1\times 1}$$

Aquest resultat correspon amb la sortida d'un Perceptró, que a continuació esdevé l'entrada d'un altre node en la següent capa. El senyal progressa d'esquerra a dreta, on el valor final de sortida es calcula realitzant aquest procediment per a tota la resta de nodes.

L'aprenentatge en sistemes de classificació, per tant, s'implementa a partir del processament continu d'exemples a través de la xarxa neuronal. Aquest conjunt d'exemples formen les dades d'Entrenament o Train. Per a cada exemple d'entrada (compost per un vector de característiques), el sistema genera una sortida a partir de les operacions internes. En tractar-se d'una funció lineal, els valors d'entrada són operats amb una matriu, on els valors del vector de sortida, anomenats puntuacions o *logits*, representen la similitud amb cada classe del sistema.

3.1.1 Funcions d'activació

En Deep Learning, podem definir el resultat d'un node o l'entrada d'un sol node o d'un conjunt a partir d'una nova funció f, la qual permet transformar el vector de resultats en probabilitats, amb valors d'entre 0 i 1, i on la suma total és igual a 1. Aquest vector final permet determinar la classe correcta a partir del valor més elevat. Normalment s'utilitza la funció de Softmax o Sigmoid, la qual és equivalent a Softmax per al cas on el nombre de classes és igual a 2.

$$y = S(z_j) = \frac{e^{z_j}}{\sum_{j=0} e^{z_j}} \tag{3.2}$$

Existeixen moltes alternatives i variants, però la funció d'activació utilitzada en el projecte s'anomena ReLU (*Rectified Linear Unit*, en anglès), que permet augmentar les propietats no lineals de la xarxa. Es defineix de la següent forma:

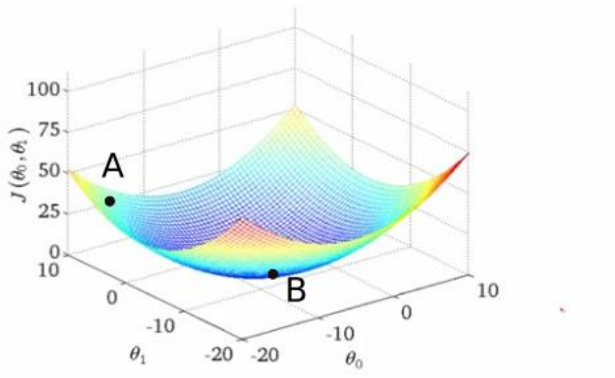
$$y = S(z) = \begin{cases} 0.01z & per\ z < 0 \\ z & per\ z \geq 0 \end{cases} \tag{3.3}$$

On z representa el senyal de sortida d'un node.

3.1.2 Optimització

Per a realitzar l'aprenentatge és necessària la computació i minimització d'una funció que mesuri l'error entre les puntuacions de sortida, i les puntuacions desitjades. El Machine Learning permet ajustar els paràmetres interns de la xarxa: els pesos i *bias*, a partir de l'optimització de la funció d'error. L'objectiu és el de navegar l'espai dels possibles conjunts de paràmetres per tal de trobar un mínim que generalitzi bé per a tot el conjunt de dades.

En un exemple idíl·lic, on la xarxa està composta únicament per dos paràmetres, s'inicialitzen els paràmetres de forma aleatòria en el punt A (vegeu *Il·lustració 3.2*), per exemple. L'objectiu és el d'obtenir una combinació de paràmetres propera al mínim B.



Il·lustració 3.2: Contorn d'una funció d'error d'una ANN formada per dos paràmetres. (Kathuria, 2018).

Un cop realitzat aquest procés d'aprenentatge, es mesura el rendiment del sistema a partir d'un nou conjunt de dades anomenat Prova o Test, el qual permet avaluar el rendiment sobre noves entrades. Normalment, aquest conjunt de Test està format pel 20% o 30% de les dades de Train.

Abans d'entrar més en detall en els mètodes utilitzats, és necessari definir certs conceptes bàsics. Primerament, és necessari explicar que les dades no s'introdueixen totes alhora dins de la xarxa neuronal, sinó que es divideixen en *batch* o conjunts. La mida d'aquests conjunts s'anomena *batch size*, i està format per un nombre d'exemples d'entrenament que passen a través de la xarxa. A continuació, és necessari definir el concepte d'*epoch*, que consisteix en tota una passada de les dades. Per a completar tot un epoch, cal realitzar un cert nombre d'iteracions sobre els conjunts de batch. Finalment, el concepte de funció d'error, es defineix com la diferència entre la predicció, és a dir, la sortida de la xarxa, amb l'objectiu a obtenir (Goodfellow, Bengio and Courville, 2016) (Dorca Saez, 2018).

Per a realitzar l'entrenament, per a cada entrada x, corresponent a una classe, es computa el conjunt de pesos w que representa millor aquella classe. L'estimació dels pesos es realitza a través de l'algoritme de gradient descendent, que permet minimitzar la funció d'error $L(w)$. Per a la computació de la gran quantitat de pesos, a partir de tots els exemples de la base de dades, es calcula un vector que indica la variació de l'error, a partir de la modificació d'un pes en una certa quantitat, anomenada *step*. El ritme que pren l'algoritme al buscar el mínim depen de l'*step*, i a aquest ritme se l'anomena *learning rate* o ritme d'aprenentatge η. Un *learning rate* massa alt pot impedir el descobriment del mínim. A continuació, s'actualitza el valor del conjunt de pesos w per tal d'aprimar-lo al mínim. Aquest procés es repeteix en cada iteració.

$$w = w - \eta \nabla_w \sum_{1}^{m} L_m\,(w) \tag{3.4}$$

A partir d'aquest algoritme bàsic han sorgit altres variants més adients per a la gran quantitat de paràmetres i d'exemples a processar. Un exemple és l'algoritme de gradient descendent estocàstic (SGD), el qual permet realitzar el càlcul de la funció d'error a partir d'un conjunt d'exemples, i no sobre totes les dades, millorant així el rendiment general (Karim, 2018). Aquest algoritme és aprofitat per un altre variant, anomenada Adam, que és l'utilitzat en el nostre sistema.

A diferència del SGD, l'algoritme d'Adam utilitza un ritme d'aprenentatge variable, i també les derivades parcials de cada paràmetre, de forma iterativa, per tal de localitzar el mínim de la funció. Ens referim a les derivades parcials d'un paràmetre x, com als valors del gradient en diferents localitzacions de l'espai de paràmetres, que equival a la derivada parcial de la funció d'error L respecte de x: $\frac{\partial L}{\partial x}$.

Adam combina els avantatges de dos algoritmes basats en el gradient descendent estocàstic: primerament l'Algoritme de Gradient Adaptatiu o AdaGrad, que manté un ritme d'aprenentatge individual per a cada paràmetre, i que millora el rendiment de problemes amb gradients dispersos (p. ex. llenguatge natural, visió per computador, ASC); en segon lloc, l'Algoritme de Propagació Quadrada de la Mitjana de les Arrels o RMSProp, que adapta el ritme d'aprenentatge dels paràmetres basant-se en la mitjana de les magnituds recents del gradient, donant resultats molt positius per a sistemes no estacionaris. Per a més informació sobre l'algoritme d'Adam veure (Brownlee, 2017) i (Kingma and Ba, 2014).

La xarxa neuronal propaga el senyal d'entrada a través dels paràmetres, fins al final de la d'aquesta, on s'avalua el comportament del model. A continuació s'aplica l'algoritme de

Backpropagation, que permet retrocedir amb la informació de l'error a través dels paràmetres de la xarxa, i d'aquesta manera alterar els paràmetres un a un.

3.2 Xarxa Neuronal Profunda (DNN)

Fins ara s'ha realitzat un recorregut bàsic per a realitzar una classificació a través d'una ANN, però existèixen altres arquitectures molt més eficients i complexes.

L'arquitectura de Deep Learning més utilitzada és la Xarxa Neuronal Profunda (Deep Neural Network, en anglès) o DNN, que pretén incrementar la profunditat del model i el nivell d'abstracció a partir d'augmentar el nombre de capes intermèdies. Aquesta estructura permet la representació de grans conjunts de dades.

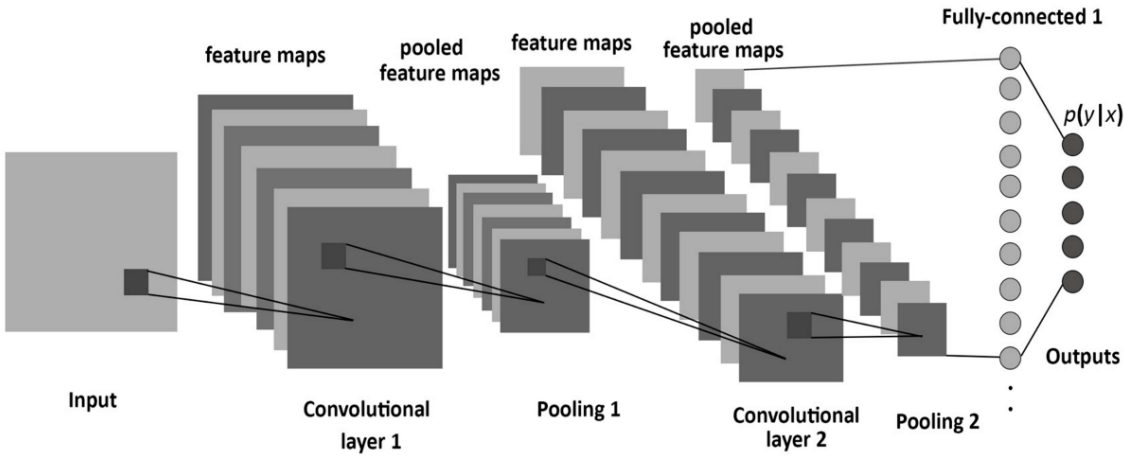
Tant l'ANN com la DNN estan formades per estructures on el flux de dades progressa des de l'entrada fins la sortida, en una sola direcció. Però per a l'estudi d'imatges i àudio, existeixen arquitectures capaces de recórrer la xarxa en diverses direccions, que permeten tenir en compte les dades prèvies, com en el cas de les Xarxes Neuronals Recurrents (Recurrent Neural Network, en anglès) o RNN, i les Xarxes Neuronals Convolucionals (Convolutional Neural Network, en anglès) o CNN.

3.2.1 Xarxa Neuronal Convolucional (CNN)

La Xarxa Neuronal Convolucional (ConvNet o CNN) deriva de l'ANN (3.1), amb la principal diferència que les dades d'entrada prenen l'estructura d'una matriu de $M \times N$ elements. És necessari, amb aquesta estructura, la implementació d'una funció general i eficient, la qual, permet reduir en gran mesura el nombre de paràmetres a entrenar. Això és realitza partir de la correlació espacial local de la matriu d'entrada, i aplicant un patró de connectivitat més escàs entre les neurones de capes adjacents.

Originalment desenvolupada per al reconeixement i classificació d'imatges, la CNN ha esdevingut recentment una de les eines més importants per a l'ASC. Van ser desenvolupades, principalment, degut a la baixa escalabilitat de les ANN per al tractament d'imatges (p. ex. donada una imatge de 32×32×3, tindria 3072 pesos únicament per a una xarxa d'una sola capa intermèdia), que desencadena molt rapidament en el sobreentrenament de les dades. Contrari a l'ANN, la CNN aprofita l'estructura matricial de l'entrada per a la construcció de capes on les neurones es distribuèixen en 3 dimensions: altura, amplada i profunditat. Cal aclarir, que el terme "profunditat" és referent a les dimensions del volum d'activació, i no a la profunditat d'una NN, que representa el nombre total de capes de la xarxa.

L'arquitectura CNN serà utilitzada en el projecte.



Il·lustració 3.3: Arquitectura CNN d'exemple (Nigam, 2018).

La CNN està formada per una seqüència de capes (*Il·lustració 3.3*), on cada una d'elles transforma el volum d'activació a partir d'una funció diferenciable. Les capes intermèdies normalment estan formades per capes Convolucionals, Pooling o reducció de mostreig, Fully-connected i normalització.

Per a la capa de Convolució, l'operació, que en aquest cas és en 2D, es realitza sobre dos senyals, la d'entrada f, i el senyal de filtratge w, anomenada *kernel*. El resultat de l'operació genera un nou senyal g. L'operació no és res més que un producte escalar:

$$g(x,y) = w * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t) \qquad (3.5)$$

Cada element del *kernel* w es considera $-a \leq s \leq a$ i $-b \leq t \leq b$.

És bastant freqüent l'inserció d'una capa de Pooling entre les successives capes Convolucionals. Aquest procés permet reduir la mida de la representació, així com el nombre de paràmetres i la computació de la xarxa, a la vegada que controlar el sobreentrenament. En aquesta capa es du a terme un procés de discretització de les mostres. El procediment és similar al de la capa de convolució, on es realitza una operació sobre una regió del senyal, però en aquest cas podem obtenir el valor màxim, amb Max Pooling, o el valor mínim, amb Min Pooling.

També és usual, l'inserció de capes de Normalització i Dropout. Igual que a l'inici de l'arquitectura se sol realitzar una normalització dels paràmetres d'entrada per tal de millorar l'eficiència de l'entrenament, també se sol introduir aquest element entre les capes intermèdies, permetent un ajust i escalament de les activacions. Per altra banda, l'utilització de capes de Dropout al final de les CNNs, permet una reducció dels nodes amb una probabilitat $1-p$.

Finalment, després de vàries xarxes CNN, el raonament d'alt nivell a la xarxa es realitza a través de les capes Fully-Connected, també anomenades Dense Layers. Les neurones d'aquesta capa tenen connexions a totes les funcions d'activació de la capa anterior, com en el cas de les ANN (3.1).

Per a més informació vegeu (Goodfellow, Bengio and Courville, 2016).

3.3 Classificació d'Escenes Acústiques (ASC)

El reconeixement i classificació d'ambients acústics a través de l'enregistrament del so es coneix amb el nom de Classificació d'Escenes Acústiques (ASC, *Acoustic Scene Classification*). Tot i el gran desenvolupament d'aquests sistemes, avui en dia, encara no hi ha cap algoritme que pugui replicar el HAS (sistema auditiu humà). L'ASC intenta resoldre un problema complex, a causa de la gran varietat de sons individuals que ocorren al mateix temps, i on sols una petita part dóna informació rellevant sobre l'escena. Els primers sistemes es van inspirar en el reconeixement de veu; paràmetres com els MFCC (4.2.1.1) han sigut utilitzats àmpliament i són establerts, en molts casos, com a sistema base per a l'ASC. També s'utilitzen característiques com la tassa d'encreuaments per zero, *spectral centroid*, *spectral roll-off*, energia de les bandes, etc. Aquestes característiques extretes de l'escena, seran l'entrada de l'arquitectura neuronal, oferint una representació abstracte del so (Virtanen, Plumbley and Ellis, 2017).

En aquest projecte s'utilitzarà el model de càlcul dels MFCC a partir d'enfinestrament, que és utilitzat àmpliament en el reconeixement automàtic de veu i àudio en general. A partir d'aquest model és possible realitzar un estudi sobre l'estacionarietat de les escenes acústiques, obtenint resultats sobre el millor model a utilitzar. És necessari, per tant, reconèixer el tipus d'activitat a classificar i diferenciar les escenes interiors de les exteriors.

Capítol 4

Sistema Baseline

El sistema Base o sistema Baseline, és el codi proveït per al repte de DCASE 2018 i que forma l'arquitectura del classificador. Proporciona un nivell d'entrada simple però relativament proper a l'estat de l'art de l'ASC. Aquest sistema disposa de totes les funcionalitats necessàries per al tractament de les dades, així com els sistemes per guardar i accedir als models, i avaluar els resultats.

Durant el procés d'enregistrament, les dades van ser mesurades simultàniament utilitzant vectors de 4 micròfons (nodes). Per tant, cada activitat va ser mesurada tantes vegades, com el nombre de micròfons que hi havia. El sistema Baseline entrena un únic model classificador, el qual utilitza cada canal com a una entrada. Per aquest motiu, cada gravació paral·lela es considera com un exemple individual durant l'entrenament. El sistema d'entrenament es basa en una arquitectura de Xarxa Neuronal, composta per Xarxes Neuronals Convolucionals (CNN) i Dense Layers, també anomenades Fully-Connected. Com a entrada a aquesta arquitectura, es proporciona un processament previ per a l'extracció de les energies dels coeficients de mel logarítmics, realitzat sobre cada un dels 4 canals de cada node, per separat. En la fase de predicció, es realitza una mitjana de cada un dels 4 resultats obtinguts per a cada node, obtenint així una única sortida per a cada vector de micròfons.

Per tant, l'objectiu del sistema consisteix en la classificació de segments d'àudio multicanal, en una de les 9 classes preestablertes, les quals representen activitats realitzades diàriament en l'entorn de la llar.

4.1 Programari i metodologia

El codi implementat en el projecte prové del sistema Baseline, de codi lliure, que ha sigut facilitat pels coordinadors de la competició de la Tasca 5 de DCASE 2018, Gert Dekkers, Peter Karsmakers i Lode Vuegen (Dekkers *et al.*, 2018). Desenvolupat per competidors d'altres edicions, el codi de sistema implementa l'ASC a través de tecnologies pioneres en aquest camp, molt properes a l'estat de l'art actual. El codi està desenvolupat amb Python 3, i sobre diverses llibreries de codi lliure com a requeriments (vegeu *Taula 4.1*). L'execució s'ha realitzat sobre el software Anaconda, que ofereix un entorn de desenvolupament senzill per a l'execució de sistemes amb grans volums de dades, anàlisi predictiu i càlculs científics, a través de més de 250 paquets. El procés s'ha executat sobre els servidors de VEU, del grup TSC (Departament de Teoria del Senyal i Comunicacions) de l'ESEIAAT, de la UPC, a través de GPUs i CPUs (TALP).

Llibreria	Versió mínima	Descripció
Dcase Util	0.2.4	Implementació d'utilitats relacionades amb bases de dades d'àudio, a través de l'ús de metadades i d'una API estandarditzada per a la manipulació d'aquestes dades (contenidors, característiques, fitxers, etc.).
Keras	2.1.5	API d'alt nivell per a la implementació de xarxes neuronals, capaç d'executar-se sobre TensorFlow. Permet una ràpida implementació.
TensorFlow	1.4	Plataforma per a la implementació del procés complet de Machine Learning a través de tensors. Facilita la implementació sobre diverses GPUs i CPUs. En el projecte s'utilitza com a *back-end* de Keras.
NumPy	1.9.2	Paquet fonamental per a la computació científica. Conté objectes vectorials de N-dimensions, funcions sofisticades, àlgebra lineal, transformades de Fourier, funcionalitats amb nombres aleatoris, etc.
Scikit-learn	0.19.1	Llibreria que permet la implementació d'algoritmes de classificació, regressió i anàlisis de grups.

Taula 4.1:Llibreries, amb les corresponents versions, necessàries per a l'execució del sistema Baseline.

El procés d'investigació i execució del sistema Baseline s'ha dividit en tres fases:

Primerament, es realitza un estudi bàsic sobre l'estructura dels 4 canals de cada àudio, analitzant les característiques bàsiques extretes pel sistema (4.2.1), que seran l'input de la xarxa neuronal. En aquesta primera fase, també es realitza una anàlisi primitiu d'escolta dels àudios, per tal de reconèixer la periodicitat i estacionalitat del senyal, per a cada una de les 9 categories a classificar. L'estudi es realitza sobre una versió reduïda de la base de dades SINS, que anomenem BDmicro, formada per l'extracció d'un 0.1%, corresponent a 72 àudios, dels 72.983 àudios totals. El total de la base de dades és anomenada BDfull, corresponent a una mida aproximada de 89 GB.

A continuació, s'ha realitzat la implementació del sistema Baseline sobre un entorn local per tal d'analitzar el codi bàsic i l'estructura dels fitxers. A partir de la configuració dels paràmetres bàsics d'execució es realitza la primera implementació del sistema, sobre una base de dades reduïda, anomenada BDmini. Aquesta base de dades està formada per l'1% de la BDfull. L'execució del programa ens permet realitzar un estudi bàsic sobre el procés d'extracció de característiques (4.2.1) i la normalització d'aquestes (4.2.2). L'execució de l'aprenentatge, però, no ha sigut possible a causa de la reduïda mida de la BDmini. A causa d'aquesta limitació, s'ha pres la decisió d'executar el sistema amb la base de dades completa.

Finalment, s'ha implementat el sistema sobre la BDfull, i l'execució s'ha realitzat en el servidor del TSC amb les característiques esmentades anteriorment. Els resultats seran exposats més endavant (Capítol 5), així com les modificacions del sistema (4.3).

Tant l'arquitectura del sistema, com l'anàlisi de la configuració dels paràmetres necessaris, seran analitzats amb més detall en els pròxims apartats.

4.2 Definició del sistema

A continuació es presenten les característiques bàsiques de l'arquitectura del sistema Baseline, així com els paràmetres per defecte.

L'estructura del sistema Baseline inclou segments d'àudios multicanal gravats per 4 nodes diferents, així com les equivalents etiquetes (*ground truth*, en anglès) i la divisió en *folds* del corresponent 4 Fold Cross-Validation'. Els Cross-Validation Folds permeten un desenvolupament i avaluació uniforme dels resultats obtinguts sobre la base de dades, a la vegada que eviten l'avaluació de models sobreentrenats. En aquest projecte s'hi implementa una divisió en k folds, on k pren el valor de 4. Per a crear aquestes divisions és necessari l'execució de varius passos. Primerament es mesclen les dades i es divideixen en k subgrups, mantenint junts els segments que pertanyen a una mateixa sessió d'una activitat (p. ex. una sessió de cuina gravada per diversos nodes), i així evitar el filtratge entre *folds*. A continuació, es selecciona un subgrup per a crear les dades de test, les quals permeten fer l'avaluació del sistema un cop entrenat; els altres $k - 1$ grups són seleccionats per a realitzar l'entrenament del sistema. Aquesta assignació es realitza de forma iterativa k vegades fins que cada un dels k subgrups ha pres el rol de test. A la vegada, se selecciona un 30% de les dades d'entrenament per a realitzar la validació del sistema cada 10 *epochs*.

L'arquitectura de *k-fold cross validation* permet extreure el màxim partit a les dades, reduint l'impacte del *bias* i estimant un valor menys optimista, però més proper a la realitat del sistema per a l'entrada de dades externes. Cada *fold*, per tant, conté els fitxers de Train (amb $3/4$ de les dades etiquetades), un altre conjunt test (amb $1/4$ de les dades sense etiquetar) i finalment, el conjunt d'avaluació (amb el mateix $1/4$ de les dades de test, però etiquetades amb el *groundtruth*).

Per a l'obtenció de les dades d'entrada a la xarxa neuronal, és necessari realitzar l'extracció de característiques (4.2.1). En aquesta primera fase es processen els segments de 10 segons a partir de finestres de 40 ms i una superposició del 50%. Les característiques resultants de cada canal del segment prenen unes dimensions matricials de 40×501 a l'entrada de la xarxa neuronal. Aquests valors corresponen als 40 MFCC de cada una de les 501 trames.

L'arquitectura de la xarxa neuronal pren cada canal del segment com a un element separat durant l'entrenament; en la fase d'avaluació el resultat general d'un segment s'obté a la sortida de l'arquitectura neuronal, a partir de la mitjana de probabilitats dels diferents canals.

L'arquitectura de la xarxa pren la següent estructura:

Capa	Característiques
CNN 1	Capa Convolucional + Normalització de Batch + Activació ReLU
	Max Pooling + Dropout
CNN 2	Capa Convolucional + Normalització de Batch + Activació ReLU
	Max Pooling Global + Dropout
Fully connected	Fully-Connected + Activació ReLU + Dropout
Sortida	Softmax

Taula 4.2: Arquitectura de la xarxa neuronal del sistema Baseline

Prèviament a l'entrada de la xarxa, se selecciona cada conjunt de validació per a cada *fold*, i se li aplica un balanceig per tal d'igualar la mida de les classes a la mida de la classe més petita.

Un cop dins de la xarxa, l'aprenentatge del model es realitza a través de l'algoritme d'Adam (3.2.2), amb un ritme d'aprenentatge η de 0,0001, i amb una mida de 256 *batch* per cada un dels 4 canals, tenint en compte, però, que cada un dels canals és processat de forma separada i que el càlcul de la mitjana dels 4 canals es realitza al final. El sistema realitza un total de 500 *epochs*, valor que permet obtenir un total de 50 models per a cada *fold*, ja que, com s'ha esmentat anteriorment, es realitza una validació per cada 10 *epochs*. Finalment, la puntuació del resultat es basa en la Puntuació F1 Macro-averaged o Macro-averaged F1-score (vegeu 4.2.5). Aquest valor és calculat per a cada classe i promitjat sobre totes elles. El model seleccionat és aquell amb el valor més elevat de la 'Macro-averaged F1-score'.

4.2.1 Extracció de característiques

Per a poder realitzar una classificació eficaç sobre les diferents categories, és necessari realitzar una extracció i processament adient de les característiques, extraient així una representació matemàtica de les dades del senyal d'àudio. En comptes de processar cada segment de 10 segons cada vegada (p. ex. $6{,}4 * 10^3$ valors dimensionals per a un segment de 10 segons, de 4 canals i a 16 kHz), el sistema primerament divideix cada segment en trames, seccions de mida inferior, de mida fixa i superposats, que permeten realitzar una divisió dels 4 canals.

Tot i que aquesta segmentació és necessària per a la realització de transformacions, tant per a reduir la variabilitat de les dades d'entrada, com per produir característiques útils o obtenir eficàcia computacional, una millor justificació per a l'extracció de característiques en segments, és que els humans (dels quals intentem capturar la percepció del HAS) també processen l'àudio en petits extractes; mentre les escenes a classificar són senyals en el temps, no es pot percebre tota una escena de cop (Schlüter, 2011).

Tot i que qualsevol segment d'àudio pot ser descrit pel conjunt de característiques locals, la majoria d'informació és redundant o irrellevant per a operacions similars. Per tant, és més útil capturar aspectes rellevant del segment i descartar detalls innecessaris a partir de l'abstracció de les característiques actuals. Models estadístics explicats a continuació (4.2.1.1), permeten realitzar aquestes transformacions.

4.2.1.1 Coeficients Cepstrals en les Freqüències de Mel (MFCC)

Els MFCC són considerats un estàndard en el camp del reconeixement d'estructures musicals i en el modelatge de contingut freqüencial i tonal de senyals d'àudio. Treballs recents en el camp de l'ASC inspirats per sistemes de reconixement de veu, han establert característiques com els MFCC com a sistemes base per al desenvolupament de l'ASC.

Es deifineix l'escala mel amb la següent formula:

$$\mathrm{F_{mel}} = \frac{1000}{\log(2)}\left[1 + \frac{\mathrm{F_{Hz}}}{1000}\right] \tag{4.1}$$

On F_{mel} és l'escala logarítmica de l'escala freqüencial normal representada per F_{Hz}.

A continuació es descriuran els passos computacionals necessaris per extreure els MFCCs, a la vegada que la motivació en el reconeixement d'escenes acústiques.

Segmentació en trames: Primerament, de cada segment de 10 segons, s'extreuen fragments de 40 ms amb una superposició del 50%. L'objectiu consisteix a obtenir extractes suficientment curts com per poder-los considerar estacionaris. Aquest procés permet obtenir una distribució freqüencial constant, ja que la informació temporal dels fragments serà perduda més endavant. Amb aquest procés es pot dividir el segment de 10 segons en 501 trames.

És usual, en senyals de veu, utilitzar fragments d'entre 10 i 20 ms (Vij *et al.*, 2016), però en les característiques espectrals d'escenes acústiques no es perceben canvis significatius en fragments d'aquest tamany tant petit. Per aquesta raó el sistema base parteix de 40 ms, com a mida de trama.

Enfinestrament de Hamming Asimètric: Cada trama es multiplica per una finestra de Hamming per tal de minimitzar la pèrdua espectral d'energia sobre els contenidors veïns, així com reduir les discontinuïtats als extrems de l'enfinestrament que poden generar components no desitjades d'alta freqüència.

Transformada Ràpida de Fourier (FFT): Una FFT és aplicada a cada trama per tal d'obtenir la distribució d'energia en freqüències, així com la fase. Es realitza la FFT amb 1024 mostres.

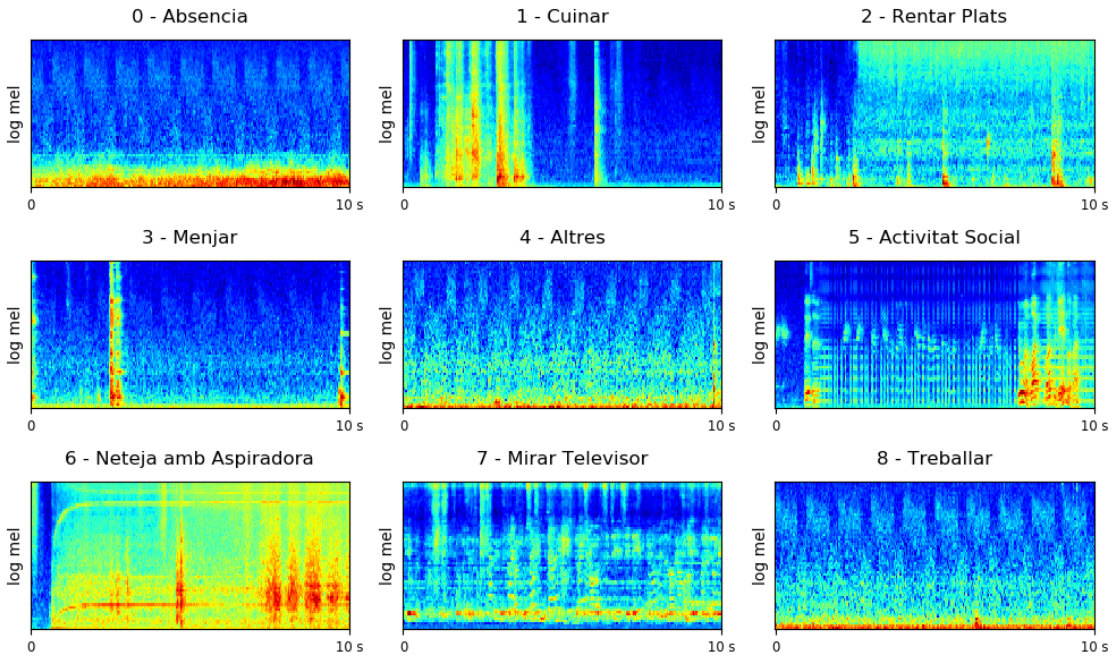
Espectre de Magnitud: La informació de la fase és descartada al prendre el valor absolut de cada valor complex del contenidor freqüencial. Es pot considerar que el HAS és "sord a la fase" sota certes circumstàncies (Gold and Morgan Nelson, 1999), indicant que la magnitud és més important que la fase.

Escala Mel: Les freqüències són mapejades a una escala perceptiva del to, l'Escala Mel, en els intervals d'entre 50 Hz, per a la freqüència inferior, i 8000 Hz, per a la freqüència superior. En estudis de reconeixement de veu, rangs de freqüència similars s'han demostrat resultats molt positiu (Ghodasara *et al.*, 2016), i al ser aquesta la principal base dels ASC, el sistema base també en fa ús. La percepció del to és percebuda pel HAS com a lineal fins als 1000 Hz. Al voltant d'aquest valor, i en freqüències superiors es percep de forma logarítmica. Aquest procés redueix la resolució de les freqüències més elevades, les quals són menys rellevants per al reconeixement d'escenes acústiques. L'Escala Mel també és utilitzada per a la reducció de la dimensionalitat: els contenidors freqüencials de l'espectre de magnitud, de 1024 mostres, es mapejen corresponentment a 40 bandes de mel, un nombre menor, que permet treballar amb més eficàcia. Aquest procés és justificable pel fet que el HAS només és sensible a les anomenades Bandes Crítiques (Gold and Morgan Nelson, 1999).

Log: La funció logarítmica s'aplica a les magnituds de mel. Es presenten dues motivacions al darrere: primerament, la percepció del HAS pel volum és logarítmica; per altra banda, el logaritme transforma el producte de les dues parts de l'àudio (la font i el filtre) en una suma, la qual és més fàcil de separar.

Transformada Discreta de Cosinus (DCT): La DCT s'aplica a cada una de les trames amb les magnituds logarítmiques de mel per tal d'evitar la correlació entre els vectors de característiques. Considerant la DCT com la Transformada de Fourier per a senyals reals, aquesta troba periodicitats en els espectres de magnitud freqüencials logarítmics (en general el resultat s'anomenat el 'cepstrum' del senyal, com a anagrama de *spectrum*). S'extreu així, l'envolvent de les magnituds espectrals, d'on s'extreu la informació més rellevant per al classificador (Ghodasara *et al.*, 2016).

A partir d'aquest procediment podem extreure l'espectrograma logarítmic de mel de diferents exemples de cada classe. Aquesta il·lustració (*Il·lustració 4.1*) permet analitzar les característiques de cada classe i establir els paràmetres d'entrada més efectius.



Il·lustració 4.1: Espectrogrames Logarítmics dels MFCC d'exemples de cada classe.

Els àudios utilitzats per a la representació de l'espectrograma són els següents:

- 0 - DevNode1_ex11_1.wav
- 1 - DevNode1_ex43_1.wav
- 2 - DevNode1_ex56_1.wav
- 3 - DevNode1_ex66_1.wav
- 4 - DevNode1_ex175_1.wav
- 5 - DevNode1_ex197_1.wav
- 6 - DevNode1_ex218_1.wav
- 7 - DevNode1_ex227_85.wav
- 8 - DevNode1_ex236_9.wav

En resum, els MFCCs descriuen l'embolcall espectral de petites trames d'un senyal. Acabat aquest procés obtenim com a resultat una matriu. Cada matriu és la representació d'un segment de 10 s, amb dimensions de 40 × 501 mostres, on observem els 40 coeficients logarítmics de mel per cada una de les 501 trames.

4.2.2 Normalització

El procés d'estandarditzar el rang de característiques o variables independents, s'anomena normalització. Aquest procés s'implementa per dues condicions: Per una banda, a causa de l'àmplia varietat de valors que prenen les característiques extretes en l'apartat anterior (4.2.1), és necessari aplicar una normalització d'aquestes dades amb l'objectiu d'aconseguir una contribució aproximadament proporcional entre aquestes (p. ex. en l'ús de la distància euclidiana és imprescindible que els valors prenguin valors dins d'una mateixa escala per a una comparació

justa). Per altra banda, l'escalament d'aquests valors s'aplica per a una millora en el rendiment de convergència en l'algorisme del gradient descendent (Ioffe and Szegedy, 2015).

En general, s'apliquen dos mètodes diferents en funció del sistema a classificar:

Normalització: Permet escalar el rang de valors entre [0,1] o [-1,1] en funció del rang seleccionat:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad x' = \frac{x - \mu}{x_{max} - x_{min}} \tag{4.2}$$

On x correspon al valor original, x' és el valor normalitzat i μ correspon al valor mitjà.

Estandarditzación: Permet obtenir una mitjana igual a 0 i variància unitària:

$$x' = \frac{x - \mu}{\sigma} \tag{4.3}$$

On x correspon al valor original i x' és el valor normalitzat. El valor de μ correspon a la mitjana i σ és la desviació tipus (*standard deviation*, en anglès) per cada trama.

Les dues opcions tenen inconvenients en funció del sistema a classificar. En el cas d'escenes acústiques, on és molt comú l'ocurrència d'*outliers*, la normalització de les característiques escalarà les dades "normals" a un interval molt petit. A l'utilitzar l'estandarditzación, les característiques no estan limitades.

Per a més infromació sobre l'aplicació dels sistemes vegeu la llibreria Dcase Util.

4.2.3 Característiques de la xarxa neuronal

La implementació de la xarxa neuronal es realitza a partir de la llibreria Keras, la qual en aquest projecte, utilitza TensorFlow com a *back-end*. També és necessària la llibreria d'Scikit-learn per a reduir el nombre de mostres després de cada *epoch*, i d'aquesta manera balancejar les classes, establint que el nombre d'exemples de totes les classes sigui igual al nombre d'exemples de la classe més petita. A través d'aquesta estructura és relativament senzill la implementació d'arquitectures sobre el model de Keras. L'entrenament s'executa a partir de la funció '.fit_generator()', que permet operar amb grans mides de dades sense guardar amb memòria, a partir d'una funció generadora que s'executa sota la creació del model, i que genera les dades. D'aquesta manera, els resultats són guardats en un fitxer Pickle cada 10 *epochs*.

L'estructura de la xarxa neuronal està formada, a grans trets, per dos CNNs, una capa Fully-connected, i finalment, una capa de sortida Softmax. A continuació s'explicarà l'estructura i funcionalitat a partir dels termes definits per la llibreria Keras.

La primera CNN està formada per 2 conjunts de capes:

1. Primer conjunt:
   - **Conv2D**: Aquesta primera capa genera un *kernel* de convolució que es convoluciona amb la capa d'entrada per a extreure un tensor de sortides. L'entrada està formada per una matriu d'$1 \times 40 \times 501$, que correspon al nombre de canals, l'eix de dades, i l'eix temporal, amb el mateix ordre. A aquesta entrada se li aplica un total de 32 convolucions de distribució normal truncada, amb un pas d'1 i sense *padding* als marges, que equivalen a la dimensionalitat de l'espai de sortida. La mida de les convolucions és de 40×5, i d'aquesta manera prenem informació temporal de les 4 trames més properes.

- **BatchNormalization**: Aquesta capa realitza una normalització de les activacions de la capa anterior, a partir d'una transformació s'obre l'eix de les dades, que manté el valor d'activació proper al 0 i la desviació estàndard d'activació propera a 1.
- **Activació ReLU**: Finalment s'aplica una funció d'activació d'Unitat Lineal Rectificada o ReLU.

2. Segon conjunt:
   - **MaxPooling2D**: En aquest segon conjunt s'aplica Max Pooling sobre les dades espacials per tal de reduir la mida general, així com evitar el sobreentrenament. En l'eix horitzontal es realitza l'operació sobre una finestra de 5 unitats, la qual concentra tota la informació temporal de les 5 trames. Obtenim un tensor de sortida de 4D.
   - **Dropout**: Finalment una fracció de 0,2 mostres d'entrada es posen a 0 per a cada actualització de l'entrenament, que ajuda a evitar el sobreentrenament.

La segona CNN està formada per la mateixa estructura que la primera, però amb la diferència que, en la Conv2D s'aplica un total de 64 convolucions de mida 1×3, i que en el segon conjunt, en comptes d'implementar MaxPooling2D, s'aplica una capa GlobalMaxPooling2D, que realitza l'operació amb una finestra igual a la mida de l'entrada.

A continuació es computa una xarxa Fully-Connected:

- **Dense**: També anomenada Fully-Connected. Aquesta capa actua com una regular ANN totalment connectada, amb una dimensió de sortida de 64 unitats. La inicialització dels pesos es fa de forma uniforme, i com en casos anteriors, s'aplica una funció d'activació ReLU.
- **Dropout**: Finalment, com en el cas anterior, s'aplica una capa de Dropout amb les mateixes característiques.

Al final de la xarxa s'hi implementa una última capa **Dense**, amb una sortida de 9 unitats, equivalent al nombre de classes del classificador. En aquest cas, en tractar-se de la capa final, s'aplica una funció d'activació Softmax, que permet realitzar la conversió a probabilitats.

Les capes CNN han demostrat una gran eficàcia en el reconeixement de patrons simples, els quals després seran utilitzats per a formar-ne de més complexes en capes superiors. Com ja hem vist, les funcions utilitzades en el sistema per a la implementació de les CNNs, són les Conv2D de Keras, on l'entrada està formada per una matriu d'un sol canal. Aquesta estructura equival a una Conv1D, les quals són molt efectives en l'extracció de característiques de segments petits i de mida fixa, principalment en aquells casos on la posició de la característica dins del segment no és de gran rellevància, com en el cas del processament de senyals d'àudio (Ackermann, 2018).

Per a més informació sobre les capes i funcions utilitzades vegeu (Capítol 3).

4.2.4 Entrenament

Per a realitzar l'entrenament de la xarxa neuronal, el sistema utilitza l'optimitzador d'Adam (3.2.2), que conjuntament amb l'algoritme de Backpropagation permet modificar els paràmetres de la xarxa fins a minimitzar l'error general del model. Adam realitza molt eficientment l'optimització del gradient de primer ordre a partir de funcions estocàstiques. Aquest algoritme és molt directe d'implementar, requereix poca memòria i és molt útil per a sistemes amb grans bases de dades o paràmetres. A la vegada, és molt útil per a objectes no estacionaris i problemes amb alt soroll, principals avantatges per les quals s'utilitza aquest sistema en el projecte.

Com a paràmetres de configuració, el sistema Baseline estableix un ritme d'aprenentatge de 0,0001. Un valor tan baix implica que el trajecte fins a convergir al mínim serà molt costós (principalment en regions d'altiplans), però asegura la localització d'aquests mínims locals. Podem definir un nou paràmetre $\theta_1{}'$ com la diferència entre el paràmetre actual θ_1 i el producte del ritme d'aprenentatge η amb el gradient:

$$\theta_1' = \theta_1 - \eta \frac{\partial}{\partial \theta_1} L(\theta_1) \tag{4.4}$$

Per a un ritme d'aprenentatge η massa gran, el gradient descendent pot superar al mínim i no convergir, i inclús divergir, però un ritme d'aprenentatge massa petit pot ser massa lent. L'optimització d'Adam permet adaptar aquest paràmetre en cada iteració entre extrems raonables, i d'aquesta manera millorar l'eficiència en la localització del mínim. Per altra banda, l'augment del ritme d'aprenentatge permet travessar més ràpidament els punts de sella sobre altiplans.

El sistema Baseline utilitza un total de 500 *epoch* per a cada *fold*, i per cada un, el conjunt de dades d'entrenament es barregen i es redueixen les mostres per tal d'obtenir classes amb una distribució uniforme. Cada 10 *epochs*, el conjunt de validació (format pel 30% dels exemples d'entrenament) permet comprovar el rendiment del sistema. Aquesta validació permet realitzar un seguiment de l'aprenentatge (Capítol 5). La mida dels *batch* són de 256 segments, per 4 canals cada un. Tenint en compte que, durant l'entrenament, el sistema tracta de forma independent cada canal, els *batch size* són de 1024 exemples.

Per al càlcul de l'error, normalment s'utilitza la funció d'Error d'Entropia Creuada, o Cross-Entropy Loss, que permet realitzar el càlcul de la diferència de la distribució de les classes observades respecte a les probabilitats estimades de cada classe.

$$CE = -\sum_{i}^{C} t_i \log(s_i) \tag{4.5}$$

On t_i correspon als valors de referència o *groundtruth* i s_i correspon a l'estimació de probabilitat per a cada classe C.

A aquesta funció se li sol aplicar una activació prèvia de Softmax (3.2.1) $f(s)_i$, que permet computar una sortida de la CNN per al conjunt C de totes les classes. A aquesta nova funció se la sol anomenar funció d'error *Categorical Cross-Entropy*. En una classificació on les etiquetes són *one-hot*, és a dir, que només una classe és seleccionada, si considerem C_p com la classe a identificar, només hi ha un element t_p a identificar que no sigui zero:

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \tag{4.6}$$

La funció d'error del sistema és la *Sparse Categorical Cross-Entropy Loss*, que en comptes de representar el vector de *groundtruth* com a un vector de zeros amb un 1 en la classe correcta, es representa com a un enter, en el nostre cas del 0 al 8, indicant una de les 9 classes a classificar.

Aquesta configuració defineix a un total de 17.769 paràmetres, 192 dels quals no poden ser entrenats (p. ex. nombre de capes intermèdies).

4.2.5 Avaluació

El sistema Baseline entrena un únic model classificador que pren un únic canal com a entrada. Cada gravació paral·lela d'una activitat és processada de forma independent durant l'entrenament, però en la fase d'avaluació, les dades de test computen una única sortida per a cada node al calcular la mitjana dels 4 segments processats anteriorment, que equivalen als 4 canals d'entrada enregistrats per a cada micròfon.

Com s'ha explicat anteriorment (4.2), el sistema utilitza una estructura de k-Fold Cross - Validation, que permet l'avaluació del model classificador per a un conjunt de dades limitades. Per a cada un dels 4 *folds*, es realitza un entrenament del model, el qual és avaluat pel conjunt de test del corresponent *fold*. Aquesta avaluació determina un percentatge per a cada una de les 9 classes.

Per a la mesura de la precisió del sistema s'utilitza l'anomenat F_1-score. Aquest sistema considera tant la precisió p com el *recall* r:

$$p = \frac{nombre\ de\ resultats\ correctes\ positius}{nombre\ de\ resultats\ positius\ del\ classificador} \qquad (4.7)$$

$$r = \frac{nombre\ de\ resultats\ correctes\ positius}{nombre\ de\ mostres\ rellevants}$$

On "nombre de mostres rellevants" representa a totes les mostres que haurien de ser identificades com a positives. El F_1-score és la mitjana harmònica de la precisió i el *recall*:

$$F_1 = \left(\frac{r^{-1} + p^{-1}}{2}\right)^{-1} \qquad (4.8)$$

Podem observar que amb aquesta avaluació, com més elevat és F_1, millor és el resultat; per altra banda, tant si la precissió com el *recall* baixen, tot el F_1 baixa. Per tant, un model obté bons resultats si les prediccions de positius són realment positius, i si no es perden positius, els quals són predits com a negatius.

Finalment, la mesura general del model es realitza a partir de l'anomenat Macro-averaged F1-score. Aquest algoritme computa cada mètrica de forma independent per a cada classe i a continuació realitza la mitjana dels resultats, característiques adequades per a un sistema balancejat com és el cas del sistema Baseline.

4.3 Modificacions del sistema

Fins aquest punt, només s'han esmentat les característiques principals del sistema Baseline, les quals implementen un model classificador proper a l'estat de l'art. Tot i això, aquest sistema es presenta com a proposta per a la implementació de millores i estudis en l'àmbit de l'ASC.

Anteriors estudis en l'àmbit de l'ASC, com el cas de "Acoustic Scene Classification Based on Spectral Analysis and Feature-Level Channel Combination" (Vij *et al.*, 2016), originari del repte de Detecció i Classificació d'Escenes i Esdeveniments Acústics, de 2016, han demostrat grans avenços en l'estudi d'extracció de característiques dins del marc de la classificació d'escenes acústiques exteriors. En aquest document es mostra la millora del sistema Baseline, a partir de

l'augment de la mida d'enfinestrament, que per defecte és de 40 ms. Els millors resultats són obtinguts per a una finestra de 2 s i un encavalcament del 50%, però cal tenir en compte que l'objectiu d'estudi d'aquest document està focalitzat en la classificació d'escenes acústiques exteriors, on els senyals a analitzar són més estacionaris.

La implementació i l'estudi d'aquesta modificació sobre el sistema Baseline del projecte permet visualitzar l'impacte de l'enfinestrament sobre escenes acústiques interiors. El principal inconvenient d'aquesta implementació resideix en les característiques dels segments enregistrats, ja que les classes a classificar tenen característiques diferents de les escenes exteriors. Un clar exemple es mostra en la comparativa de característiques entre dues escenes: una estació de tren manté un espectrograma molt homogeni, amb unes freqüències marcades durant grans períodes de temps, per altra banda, la funció de rentar els plats té unes característiques molt diferents, amb un espectrograma més diversificat i puntual.

Aquestes diferències requereixen d'una extracció de característiques menys espaiada en el temps. La implementació de finestres de 2 s sobre àudios de 10 s no seria una bona opció, tenint en compte les característiques espectrals observades en els coeficients MFCC de cada classe (vegeu *Il·lustració 4.1*). Per aquesta raó, la implementació de finestres més petites serà necessari en el context d'aquest projecte.

En el reconeixement de veu, l'enfinestrament de segments amb mides al voltant dels 25 ms i amb encavalcaments pròxims als 10 ms o del 50% són utilitzats per a la correcta interpretació de la parla. El senyal de la veu varia massa en finestres molt grans, i per aquesta raó s'utilitzen finestres que permetin capturar els patrons de diferents fonemes. L'ASC ha aprofitat moltes característiques dels sistemes de reconeixement de la parla, els quals han sigut implementats directament sobre segments amb característiques diferents.

Les mides utilitzades seran de 500 ms i 80 ms, amb un encavalcament del 50%. A la vegada, per a la implementació d'aquest enfinestrament és necessari la configuració de la mida d'entrada de la xarxa neuronal, ja que per a un enfinestrament d'una mida més elevada en resulten menys nombre de trames. De la mateixa manera, és necessari modificar també el nombre de coeficients FFT a extreure de les trames.

Capítol 5

Resultats

Aquest capítol presenta els resultats obtinguts en l'execució dels diversos sistemes. Primerament, es presenten els resultats del sistema Baseline (5.1), on s'analitzen les principals mètriques obtingudes. A continuació, es presenten els resultats de les modificacions aplicades al sistema (5.2), així com l'impacte i la relació sobre el sistema Baseline.

L'evolució del projecte està marcada per diverses fases, com es comenta anteriorment (4.1). Inicialment, la BDmicro ens ha permès realitzar l'estudi freqüencial de les diverses escenes (4.2.1). A continuació, la implementació del sistema Baseline sobre la BDmini no ha sigut possible a causa de l'estructura del generador de model, de la llibreria Keras. La funció 'fit_generator()' requereix d'una cert mida per a la reducció de mostres, en el balanceig de les dades d'entrenament, després de cada *epoch*. Aquesta limitació desencadena en què el sistema no reconeix la totalitat de les 9 classes. A causa d'aquesta limitació la implementació del sistema es realitza directament sobre la totalitat de la base de dades, BDfull.

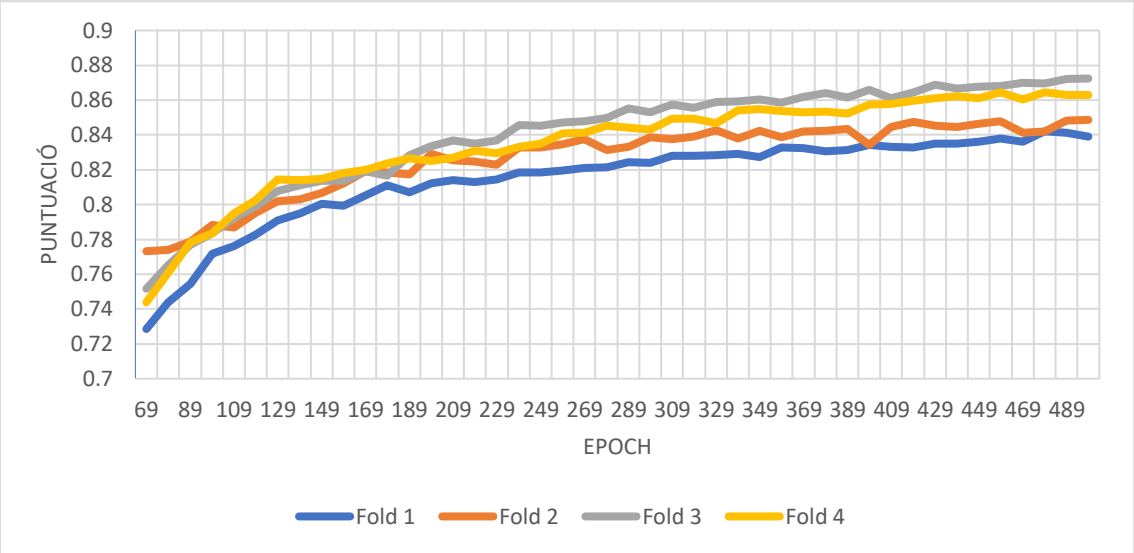
Com s'ha esmentat anteriorment (4.1), la implementació i execució del sistema s'ha realitzat en els servidors de VEU del grup de recerca TSC. En aquests servidors, l'execució de l'entrenament s'ha realitzat a través d'un script, que permet executar el sistema a partir d'un nombre de CPU/GPU i determinades GB de memòria. Cada fase del sistema (extracció de característiques, normalització, entrenament, test i avaluació), així com els paràmetres sobre els quals s'executa, són establerts prèviament en un fitxer de configuració amb extensió 'yaml'.

Finalment, ha sigut necessària la creació de sistemes automàtics que permetin l'estudi de tals quantitats de dades. Tant en els resultats mostrats a continuació, com en l'Annex (Annex A, Annex B, Annex C), podem observar les dades extretes d'aquest processament: gràfiques del progrés d'entrenament, taules amb els resultats i matrius de confusió.

5.1 Resultats del sistema Baseline

A partir dels paràmetres establerts per defecte, que s'han revisat durant la definició del sistema Baseline (Capítol 4), obtenim les mètriques i els resultats.

A continuació (*Il·lustració 5.1*, *Il·lustració 5.1*) i, amb més detall, en l'Annex A (*Il·lustració A.1*), podem observar les diferents gràfiques de l'evolució de l'entrenament de cada un dels 4 *folds*.



Il·lustració 5.1: Evolució d'aprenentatge dels 4 folds del sistema Baseline per als 500 epochs.

Com podem observar, l'evolució del conjunt de validació implementat amb l'F1-score mostra el millor estat com més elevada és la puntuació.

Per al model del sistema Baseline, obtenim els millors resultats de la validació de l'entrenament en els següents *epoch*, per a cada *fold*:

- Fold 1: 84,21%
- Fold 2: 84,86%
- Fold 3: 87,24%
- Fold 4: 86,45%

Durant l'entrenament del model, per tant, obtenim la millor validació per al *fold* 3.

Un cop realitzat l'entrenament, en la fase següent fase es processa el conjunt de test a través de la xarxa, que dóna resultats per a cada una de les classes (vegeu *Taula B.1*). Els millors resultats es presenten amb les classes de "Cuinar", "Activitat social", "Neteja amb aspiradora" i "Mirar la televisió". Des del punt de vista del HAS, podem determinar que aquestes classes són les més característiques freqüencialment (vegeu *Il·lustració 4.1*), amb energia elevada i freqüències fonamentals permanents. Per altra banda, la classe d'"Altres" resulta identificada amb un 44,60% dels casos, com a mitjana dels 4 *folds*, principalment a causa de la facilitat amb la qual el sistema confon aquesta classe. Podem observar aquesta característica en les matrius de confusió del sistema Baseline (*Taula B.2*, *Taula B.3*, *Taula B.4* i *Taula B.5* per a cada un dels *folds*), a la vegada es veu que les activitats amb les que hi ha més confussió són l'"Absència" i "Treballar". Aquestes activitats comparteixen característiques freqüencials molt similars, com podem apreciar en la representació dels coeficients de mel (*Il·lustració 4.1*), on es mostra, en general, poca energia excepte en les freqüècies baixes. La resta de classes es mantenen amb valors al voltant del 80%.

Per a cada *fold* obtenim el valor de Macro-Averaged F1-score:

- Fold 1: 84,09%
- Fold 2: 81,08%
- Fold 3: 84,93%
- Fold 4: 88,43%

El model del Baseline, per tant, identifica correctament el 86,63% (valor extret del promig del F1-score dels 4 *folds*).

5.2 Resultats de les modificacions

A partir de l'observat anteriorment en l'apartat (4.3), on es mostra l'impacte d'augment de l'enfinestrament en la classificació d'escenes acústiques exteriors, s'implementen variacions que permeten observar l'impacte d'aquestes modificacions en escenes enregistrades a l'interior (p. ex. un habitatge).

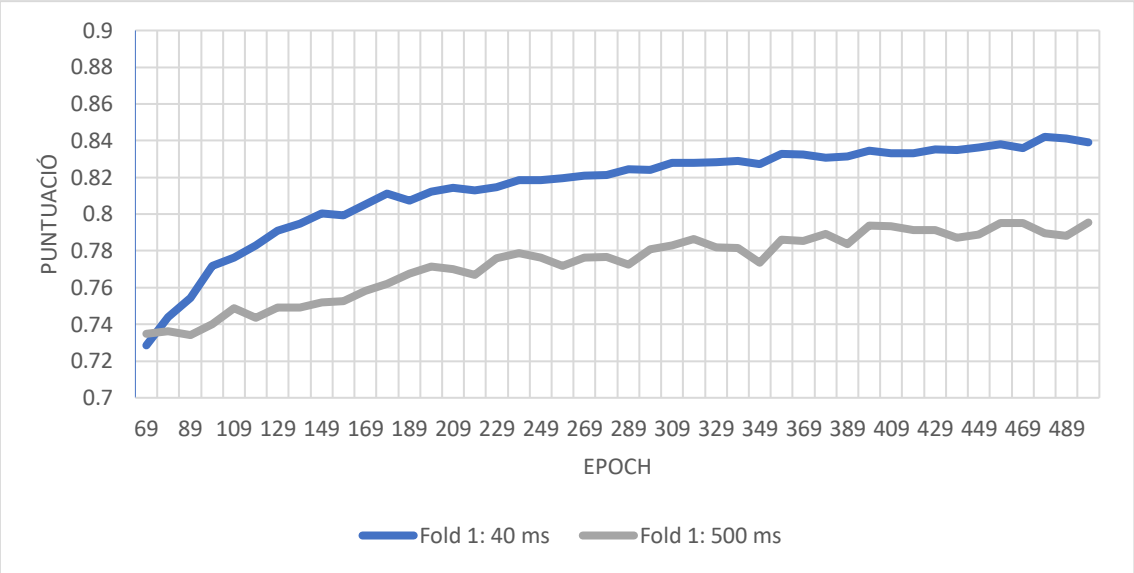
Durant el projecte, únicament s'implementen dues alteracions degut al llarg temps de computació requerit per a l'execució total del sistema (aproximadament 2,5 h per cada 10 *epochs*). Per aquesta mateixa raó, l'estudi realitzat a continuació només s'ha realitzat sobre el *fold* 1.

Per a la implementació d'aquestes modificacions és necessari realitzar vàries modificacions, no només en la fase d'extracció de característiques, sinó també en les dimensions bàsiques de la xarxa neuronal. D'aquesta manera, és possible adaptar la matriu dels coeficients MFCC, de cada segment, a les diferents capes de la xarxa neuronal.

5.2.1 Enfinestrament de 500 ms

En primera instància, i seguint el model proposat en (Vij *et al.*, 2016), on els millors resultats són obtinguts per a un enfinestrament de 2 s i 50% de encavancament, s'ha optat per a l'execució del sistema amb les següents característiques: 500 ms de llargada de finestra i 50% d'encavancament. Aquesta reducció de l'enfinestrament sorgeix a partir de l'estudi realitzat sobre les característiques freqüencials (4.2.1, 4.3), on s'observen classes molt homogènies freqüencialment, com en el cas de "Neteja amb aspiradora" o "Rentar plats", similars a les classes del sistema proposat per (Vij *et al.*, 2016). Altres classes mostren característiques totalment diferents: freqüències puntualitzades i esdeveniments sobtats, que mostren ser poc susceptibles de millora davant d'aquestes modificacions.

Per a aquesta modificació inicial els resultats no són positius, i mostren resultats molt baixos en comparació amb el sistema Baseline (vegeu *Il·lustració 5.2* i *Il·lustració A.2*).



Il·lustració 5.2: Comparació d'aprenentatge del *fold* 1 del sistema Baseline amb el *fold* 1 del sistema amb enfinestrament de 500 ms, per als 500 *epochs*.

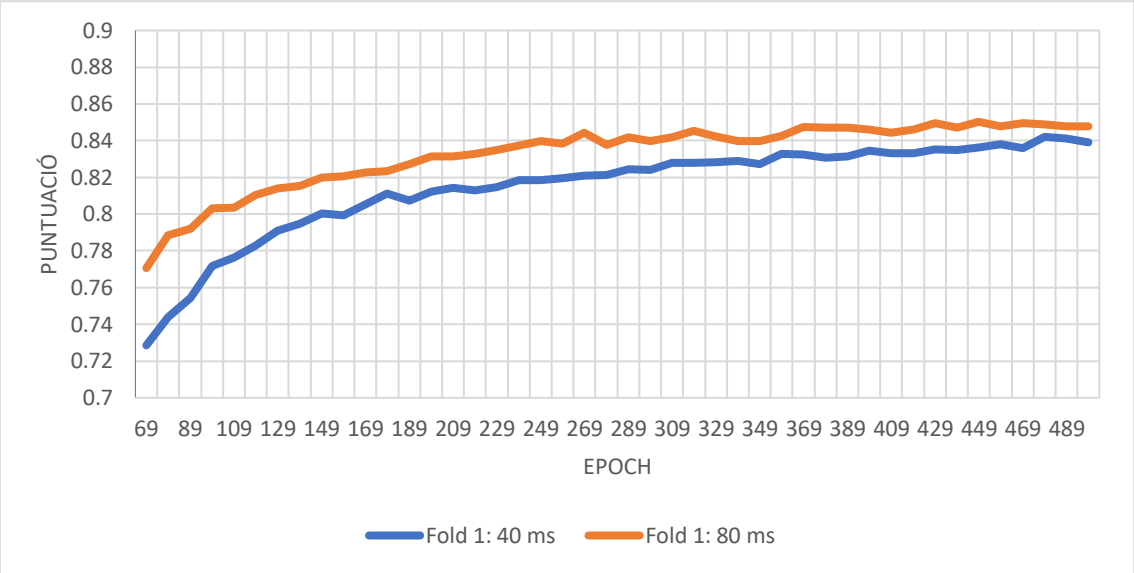
La corba d'aprenentatge mostra un progrés sobtat a l'inici, que permet augmentar la puntuació més rapidament que els altres sistemes, però comença a disminuir a partir del *epoch* 40, on la corba d'aprenentatge passa a tenir una disposició més lineal, arribant al punt màxim en l'últim *epoch*, el 499, amb un resultat de 79,54% sobre el conjunt de validació. L'envolvent d'aprenentatge mostra les característiques bàsiques d'un model amb un ritme d'aprenentatge massa elevat.

L'avaluació del sistema no mostra resultats positius en cap escena a classificar, excepte per "Neteja amb aspiradora", on gràcies a l'estacionalitat freqüencial, mostra un augment del 2,23% respecte al sistema Baseline, a l'obtenir un resultat de 99,79% (vegeu *Taula C.1*). Les classes restants mostren una disminució important en la classificació de les escenes, passant d'una mitjana general de 84,09 del sistema Baseline, a un total de 80,67%.

5.2.2 Enfinestrament de 80 ms

Finalment, i a partir dels resultats obtinguts amb els paràmetres anteriors, s'ha optat per una reducció de l'enfinestrament, més adaptat a les característiques generals de les escenes a classificar. A partir d'un enfinestrament de 80 ms i un encavalcament del 50%, el sistema és capaç de capturar la informació més singular i puntual, a la vegada que produeix trames amb informació més general sobre l'escena.

A continuació podem observar la corba d'aprenentatge del sistema, on el millor resultat es mostra en l'*epoch* 449, amb un 85,02%, respecte al sistema Baseline, on el millor resultat és de 84,20% en l'*epoch* 479 (vegeu *Taula A.1*).



Il·lustració 5.3: Comparació d'aprenentatge del *fold* 1 del sistema Baseline amb el *fold* 1 del sistema amb enfinestrament de 80 ms, per als 500 *epochs*.

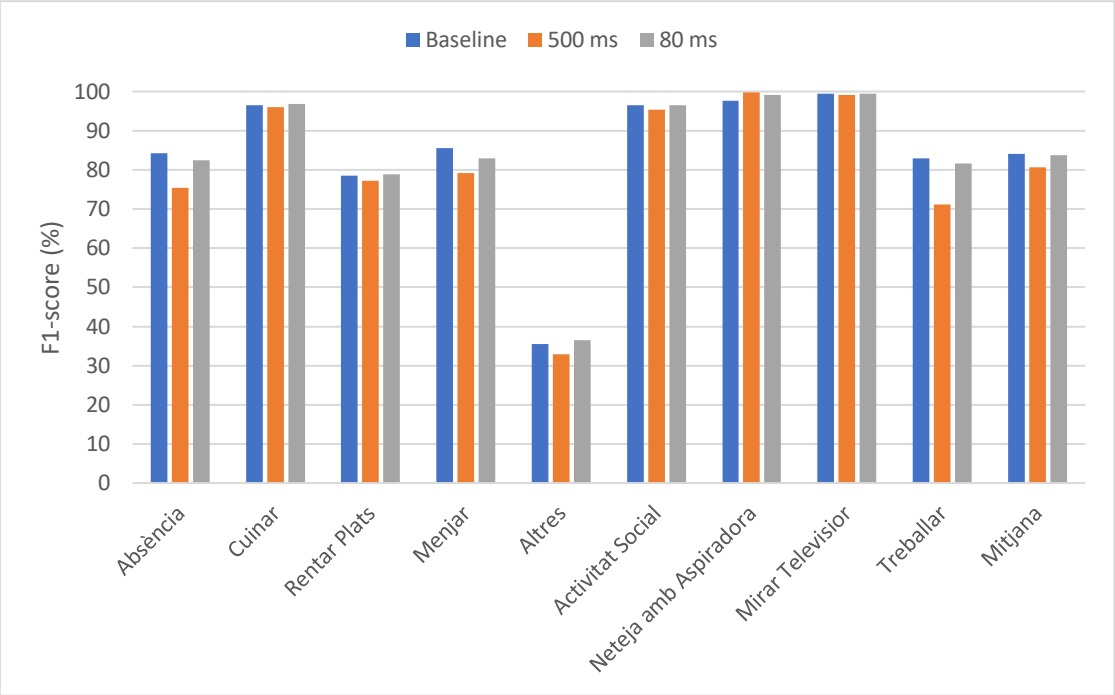
En l'avaluació del sistema, per a cada classe hi podem observar diverses característiques importants (vegeu *Taula C.2*). Per una banda observem millores en les escenes de "Cuinar", "Rentar plats", "Altres", "Activitat social", "Neteja amb aspiradora" i "Mirar televisor". Les altres escenes, però, disminueixen la certesa del sistema, principalment els casos d'"Absència" i "Menjar", que disminueixen un 1,86% i 2,65%, respectivament, i amb menys mesura l'escena de "Treballar". Aquestes disminucions disminuèixen el resultat general en un 83,81%, respecte als 84.09% del sistema Baseline.

Tot i aquesta disminució general, observem una majoria d'escenes que han augmentat la seva certesa, amb una millora en 6 de les 9 classes a classificar.

5.3 Resultats generals

A partir dels sistemes implementats extraiem tot un conjunt de matrius de confusió (Annex B, Annex C) que permeten extreure informació general de les escenes. Primerament, podem observar que la classe d'"Absència" és fàcilment confosa amb "Treballar" i "Altres". Aquest fet es deu, en gran part, a la similitud en l'espectre freqüencial (*Il·lustració 4.1*). Per altra banda, observem que les escenes de "Neteja amb aspiradora" i "Mirar televisió" tenen un espectrograma molt característic, amb una elevada energia, en el cas de "Neteja amb aspiradora", i amb freqüències de parla humana puntuals, en el cas de "Mirar televisor".

A continuació podem observar els resultats de cada una de les classes, per cada un dels tres sistemes desenvolupats:



Il·lustració 5.4: Gràfic dels resultats generals de cada classe sobre el *fold* 1.

Capítol 6

Conclusions

Aquest projecte ha sigut executat amb l'objectiu d'estudiar i implementar un sistema de detecció i classificació d'escenes acústiques domèstiques. A partir del sistema Baseline, provinent la tasca 5 de DCASE Challenge 2018, ha sigut possible la implementació d'una arquitectura formada per una extracció de característiques MFCC, i una estructura neuronal de Deep Learning, composta per xarxes CNN i capes Fully-Connected. Finalment, i a partir de l'estudi realitzat sobre el sistema, ha sigut possible l'execució de modificacions i millores sobre les característiques d'entrada de l'arquitectura neuronal artificial.

L'extracció dels paràmetres d'entrada del sistema, ha permès l'estudi de les característiques bàsiques de les 9 classes a classificar. Aquestes classes preestablertes formen part del conjunt d'activitats realitzades a la llar i que tenen com a objectiu ser identificades (p. ex. "Rentar plats", "Mirar televisor", "Netejar amb aspiradora", etc.). A la vegada, l'extracció d'aquests paràmetres ha permès la modificació de l'estructura de les dades a l'entrada de l'arquitectura de Deep Learning. Més endavant, l'estudi d'aquestes característiques ha permès la modificació del sistema Baseline, a partir de l'increment de l'enfinestrament sobre els segments d'exemple. Aquestes modificacions han demostrat ser beneficials per a 6 de les 9 classes a identificar, amb l'enfinestrament de 80 ms, principalment les escenes més particulars i estacionàries, poc susceptibles a incloure elements acústics puntuals en el temps. Inicialment, però, la implementació d'enfinestrament amb una mida de 500 ms no ha mostrat millores sobre el sistema.

La implementació del sistema Baseline sobre l'entorn d'Anaconda, és executat a partir d'instruccions de l'API de Keras, i amb TensorFlow com a element de *back-end*. Aquesta disposició, permet la implementació, en l'àmbit d'usuari, d'una arquitectura capaç de classificar escenes acústiques amb resultats al voltant del 84% d'encert. La utilització de la llibreria Dcase Util permet la implementació de classes i funcionalitats especialitzades en el processament de grans bases de dades d'àudio.

6.1 Plà de treball i modificacions

Inicialment, l'estructura temporal del projecte es dividia en 4 grans fases (explicades en 4.1). La primera consistia en l'estudi de la BDmicro, que permetia extreure informació inicial sobre les escenes a classificar. A continuació, s'implementava l'arquitectura del sistema Baseline sobre la BDmini. Aquesta primera implementació no va ser possible a causa de l'estructura interna del sistema, i per aquest motiu, la implementació del sistema Baseline sobre la BDfull en els servidors de VEU es veuria avançada. Un cop executat el sistema sobre cada un dels 4 *folds* es van realitzar les modificacions. Desafortunadament, la implementació de moltes de les modificacions sobre el sistema Baseline no han sigut possibles a causa del sorprenent temps de processament necessari per a l'entrenament del sistema. Aquesta característica es deu, principalment, a la gran mida de la base de dades.

6.2 Pressupost

L'execució dels diversos sistemes s'ha realitzat en els servidors de VEU de TSC, i per aquest motiu no ha suposat cap cost real. Una estimació del cost aproximat per hora és realitzada a partir dels serveis de Google Cloud Platform, amb un preu al voltant de 0,80€/h.

Tant el codi implementat, com les eines i llibreries utilitzades durant el projecte són de codi lliure, i per tant, no suposen cap cost real.

Per altra banda, el cost salarial dels enginyers dedicats al projecte, i a partir de la duració d'aquest (aproximadament 16 setmanes), podem establir una aproximació del total necessari:

	€ / hora	**Dedicació**	**Total**
Enginyer Junior	12 €/h	20 h/setmana	3.840 €
Enginyer Senior	30 €/h	2 h/setmana	960 €
Servidor	0,80 €/h	1.344 h	1.075 €
		Total	**5.875 €**

Taula 6.1: Pressupost general del projecte.

6.3 Futur desenvolupament

A partir de les variacions implementades sobre el sistema, podem observar el gran impacte que desenvolupa l'extracció correcta de característiques. Seguint amb aquest fil, la selecció correcta del nombre d'MFCC determina un factor clau en la representació de l'envolvent espectral del senyal. Els coeficients de baix ordre representen els aspectes més simples de la forma espectral, mentre que els coeficients de més alt ordre tenen un contingut més sorollós, i no són tan important per a l'entrenament. La correcta selecció del nombre de coeficients MFCC determina gran part dels resultats obtinguts (Ghodasara *et al.*, 2016).

Per altra banda, si es manté l'arquitectura establerta pel sistema Baseline, altres paràmetres a modificar que poden ser estudiats, de cara a millores, són aquells que intervenen en l'optimització del sistema, com la mida dels *batch*, o el percentatge d'exemples d'entrenament, respecte al conjunt de validació.

Futures direccions poden incloure la representació d'esdeveniments acústics a partir de l'estructura general de la seqüència d'esdeveniments, i la comparació del sistema Baseline amb una estructura de detecció basada en el mínim enregistrament necessari perquè el classificador identifiqui l'escena.

Altres millores implementades en el sistema requereixen la modificació del model establert pel Baseline. Altres participants en la tasca obtenen grans millores en els resultats a partir de la implementació de xarxes neuronals recursives (RNN) o màquines de vectors de suport (SVM), conjuntament amb l'estructura CNN del sistem

Bibliografia

Ackermann, N. (2018) *Introduction to 1D Convolutional Neural Networks in Keras for Time Sequences*. Available at: https://blog.goodaudience.com/introduction-to-1d-convolutional-neural-networks-in-keras-for-time-sequences-3a7ff801a2cf (Accessed: 2 June 2019).

AudioSet (no date). Available at: https://research.google.com/audioset/ (Accessed: 15 May 2019).

Brownlee, J. (2017) *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ (Accessed: 25 May 2019).

DCASE Community (2018). Available at: http://dcase.community/ (Accessed: 14 May 2019).

Dekkers, G. *et al.* (2017) 'The SINS database for detection of daily activities in a home environment using an acoustic sensor network.', 32(0).

Dekkers, G. *et al.* (2018) 'DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics'. Available at: http://dcase.community/challenge2018/task-monitoring-domestic-activities (Accessed: 21 April 2019).

Dertat, A. (2017) *Applied Deep Learning - Part 1: Artificial Neural Networks*. Available at: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6 (Accessed: 19 May 2019).

Dorca Saez, G. (2018) 'Neural Audio Generation for Speech Synthesis', (January).

File:Artificial neural network.svg - Wikimedia Commons (2011). Available at: https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg (Accessed: 19 May 2019).

Freesound - Freesound (no date). Available at: https://freesound.org/ (Accessed: 15 May 2019).

Gert Dekkers, Steven Lauwereins, Bart Thoen, Mulu Weldegebreal Adhana, Henk Brouckxon, Toon van Waterschoot, Bart Vanrumste, Marian Verhelst, and P. K. (no date) *AdvISe lab @ technology campus Geel – Ku Leuven*. Available at: https://iiw.kuleuven.be/onderzoek/advise/index_oud.html (Accessed: 14 May 2019).

Ghodasara, V. *et al.* (2016) 'Acoustic Scene Classification Using Block Based Mfcc Features', (September), pp. 2–6. doi: 10.13140/RG.2.2.18614.09287.

Gold, B. and Morgan Nelson (1999) *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. Edited by B. Zobrist. Wiley.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. MIT Press. Available at: http://www.deeplearningbook.org/ (Accessed: 14 April 2019).

Ioffe, S. and Szegedy, C. (2015) 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'. Available at: http://arxiv.org/abs/1502.03167 (Accessed: 17 May 2019).

Karim, R. (2018) *Step-by-Step Tutorial on Linear Regression with Stochastic Gradient Descent*. Available at: https://towardsdatascience.com/step-by-step-tutorial-on-linear-regression-with-stochastic-gradient-descent-1d35b088a843 (Accessed: 6 April 2019).

Kathuria, A. (2018) *Intro to optimization in deep learning: Gradient Descent*. Available at: https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/ (Accessed: 24 May 2019).

Kingma, D. P. and Ba, J. (2014) 'Adam: A Method for Stochastic Optimization'. Available at: http://arxiv.org/abs/1412.6980 (Accessed: 31 May 2019).

Nigam, V. (2018) *Understanding Neural Networks. From neuron to RNN, CNN, and Deep Learning*. Available at: https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90 (Accessed: 25 May 2019).

Schlüter, J. (2011) 'Unsupervised Audio Feature Extraction for Music Similarity Estimation', *October*.

Schmidhuber, J. (2014) 'Deep Learning in Neural Networks: An Overview'. doi: 10.1016/j.neunet.2014.09.003.

Vij, D. *et al.* (2016) 'Acoustic Scene Classification Based on Spectral Analysis and Feature-Level Channel Combination', (September), pp. 3–5.

Virtanen, T., Plumbley, M. D. and Ellis, D. (2017) *Computational analysis of sound scenes and events*, *Computational Analysis of Sound Scenes and Events*. doi: 10.1007/978-3-319-63450-0.

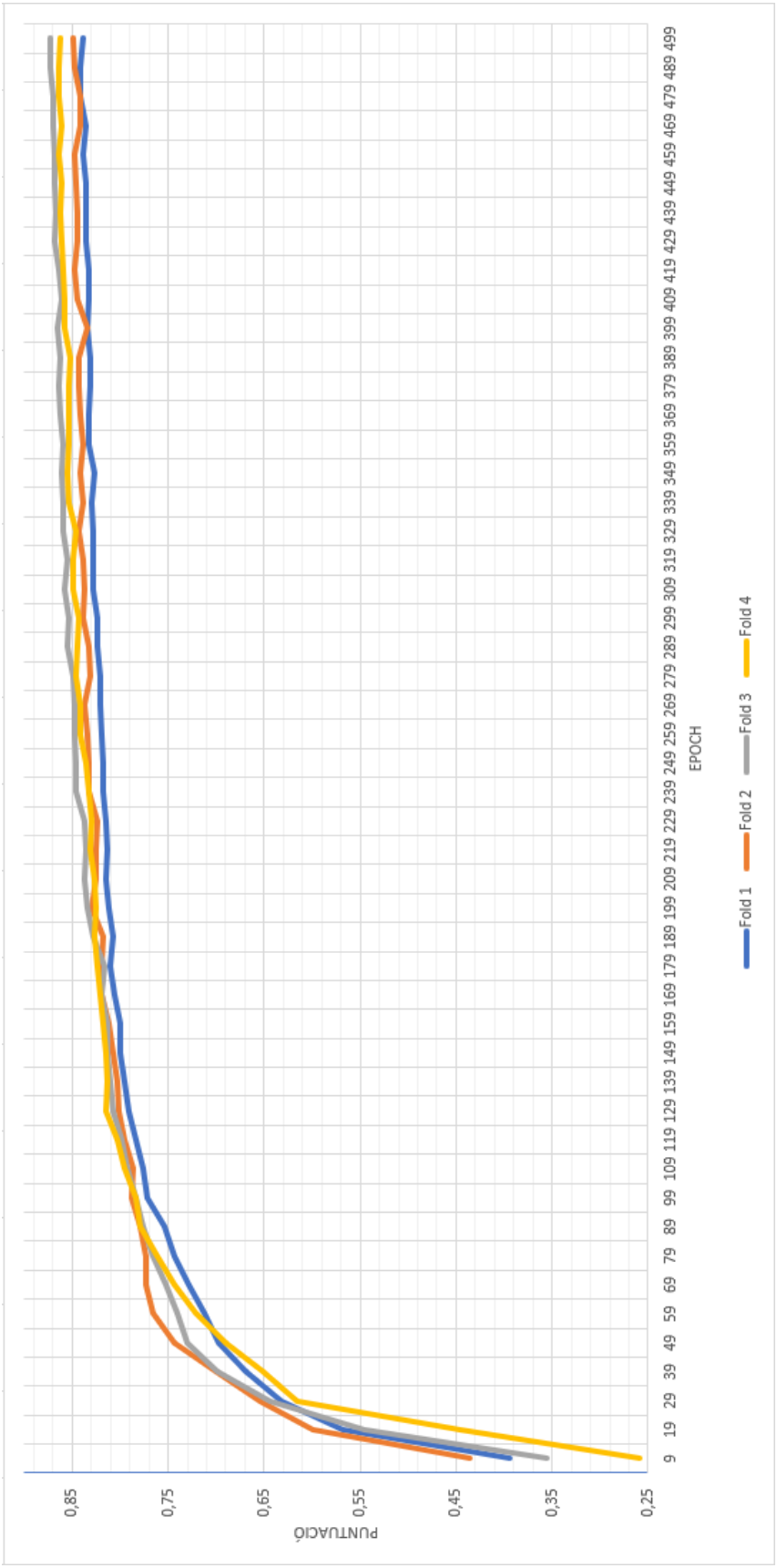
Annex A

Entrenament del sistema

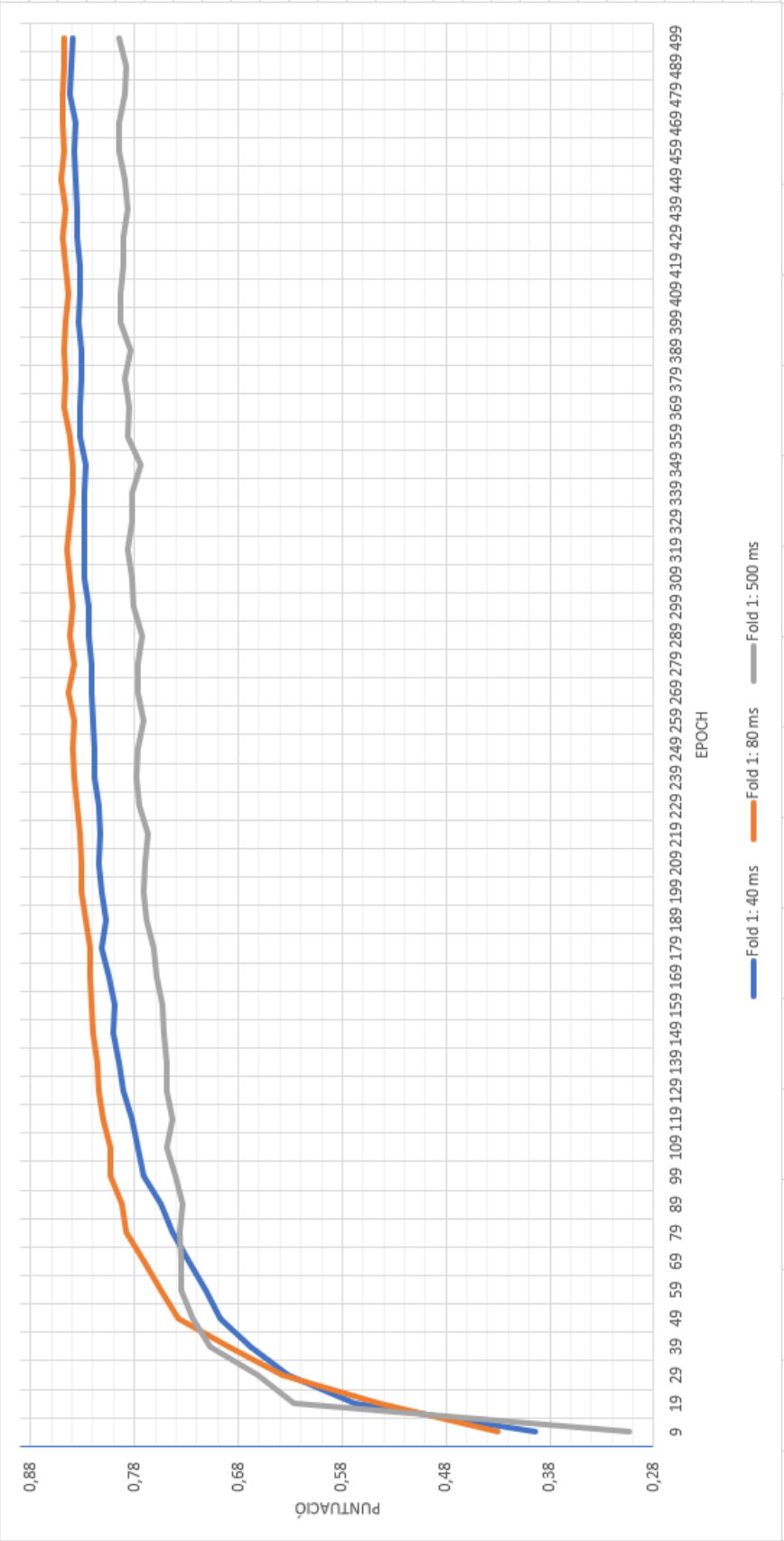
	Sistema Baseline				Millores del Sistema Baseline	
Epoch	Fold 1	Fold 2	Fold 3	Fold 4	Fold 1 (500 ms)	Fold 1 (80 ms)
9	0,394016	0,436096	0,355512	0,258705	0,303573	0,430428
19	0,568278	0,598652	0,546266	0,446074	0,627054	0,543766
29	0,63166	0,654313	0,643823	0,615854	0,661096	0,638102
39	0,668612	0,701004	0,699155	0,650759	0,706523	0,688877
49	0,696848	0,742988	0,730496	0,689927	0,723819	0,737245
59	0,711663	0,76549	0,739709	0,721813	0,735078	0,75403
69	0,728544	0,773258	0,751655	0,743883	0,734676	0,770638
79	0,743802	0,773849	0,765159	0,760735	0,736245	0,788587
89	0,754348	0,778622	0,776785	0,778495	0,734155	0,792156
99	0,771682	0,788149	0,783455	0,783669	0,740026	0,803216
109	0,776179	0,787011	0,792489	0,794798	0,748736	0,80366
119	0,782899	0,795464	0,798927	0,80264	0,743729	0,810437
129	0,790996	0,802042	0,80786	0,814368	0,749071	0,814043
139	0,794811	0,803057	0,810933	0,813915	0,74921	0,815302
149	0,800528	0,806654	0,813697	0,814619	0,751956	0,819876
159	0,799288	0,812343	0,813651	0,818054	0,752732	0,820649
169	0,805217	0,819306	0,819219	0,820016	0,758315	0,822792
179	0,811026	0,818641	0,816603	0,823775	0,762157	0,823244
189	0,807203	0,817354	0,828493	0,826535	0,767754	0,827136
199	0,81212	0,828964	0,833649	0,82491	0,771304	0,831453
209	0,814193	0,825424	0,836699	0,826949	0,770082	0,831363
219	0,812867	0,824801	0,835068	0,830844	0,767069	0,832708
229	0,814591	0,823055	0,836728	0,829442	0,775831	0,834869
239	0,818417	0,832744	0,845675	0,833155	0,77876	0,837385
249	0,818475	0,832687	0,845355	0,835163	0,776484	0,83967
259	0,819482	0,834497	0,846963	0,840888	0,7717	0,838226
269	0,821017	0,837655	0,848001	0,841292	0,776258	0,844213
279	0,821247	0,831366	0,849678	0,845373	0,776624	0,837549
289	0,824358	0,833056	0,855342	0,844224	0,772506	0,841858
299	0,824018	0,838782	0,852893	0,843069	0,780753	0,839696
309	0,828003	0,837556	0,857289	0,84936	0,78295	0,841866
319	0,827919	0,838951	0,855643	0,849355	0,786376	0,845219
329	0,828265	0,842898	0,858821	0,846621	0,781918	0,842272
339	0,828974	0,837937	0,859353	0,854099	0,781668	0,839778
349	0,827151	0,84226	0,860513	0,854777	0,773673	0,839871
359	0,832797	0,838509	0,858694	0,8539	0,786116	0,842632
369	0,832484	0,841913	0,861871	0,852874	0,785337	0,847336

379	0,830647	0,842545	0,864221	0,853354	0,789244	0,847046
389	0,831513	0,843646	0,861608	0,852238	0,783586	0,847148
399	0,834443	0,834275	0,865924	0,857528	0,793781	0,846206
409	0,833048	0,844672	0,861273	0,857862	0,793563	0,844197
419	0,832986	0,847576	0,864456	0,85949	0,791326	0,846181
429	0,835098	0,845134	0,869015	0,861047	0,791336	0,84944
439	0,834921	0,844397	0,866665	0,862277	0,787165	0,84699
449	0,836202	0,846403	0,867727	0,861107	0,788939	0,85027
459	0,837894	0,847958	0,867985	0,864473	0,795053	0,847732
469	0,83597	0,841354	0,870047	0,860361	0,795135	0,849455
479	0,842095	0,841863	0,869738	0,864499	0,789471	0,849007
489	0,841342	0,848104	0,872322	0,863117	0,788346	0,847643
499	0,839017	0,848655	0,87243	0,863023	0,795453	0,847685

Taula A.1: Progrés d'aprenentatge per cada 10 *epochs* del sistema Baseline i les millores. Els millors resultats de cada model es mostren ressaltats.



Il·lustració A.1:Progressió d'aprenentatge del sistema Baseline per a cada un dels 4 *folds*.



Il·lustració A.2: Comparació d'aprenentatge de les modificacions (80 ms i 500 ms), respecte al *fold* 1 del sistema Baseline.

Annex B

Resultats del sistema Baseline

Scene	Fold 1	Fold 2	Fold 3	Fold 4	Mitjana
Absència	84.23	86.88	89.33	90.04	87.62
Cuinar	96.52	93.67	93.25	95.87	94.83
Rentar plats	78.56	63.01	73.44	84.25	74.82
Menjar	85.54	78.19	87.68	91.46	85.72
Altres	35.43	42.68	49.43	50.84	44.60
Activitat social	96.48	87.30	92.11	96.41	93.08
Neteja amb aspiradora	97.66	99.62	100.00	100.00	99.32
Mirar televisior	99.40	99.87	97.70	99.88	99.21
Treballar	82.96	78.53	81.46	87.10	82.51
Mitjana	**84.09**	**81.08**	**84.93**	**88.43**	**84.63**

Taula B.1: Resultats del conjunt de test de cada un dels 4 *folds* i per a cada una de les classes a avaluar.

Valors de Predicció

Valors Reals	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	4269	4	4	2	311	10	0	0	132
Cuinar	0	1235	24	0	23	1	0	0	1
Plats	0	22	295	6	30	0	0	0	3
Menjar	14	0	21	485	12	0	0	3	37
Altres	157	7	27	20	239	5	0	0	65
Social	16	6	0	6	0	988	0	20	0
Aspiradora	0	1	8	0	0	0	230	1	0
Televisior	2	0	9	15	0	3	1	4545	1
Treballar	946	0	7	28	214	5	0	0	3504

(a)

Valors de Predicció

Valors Reals	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	0.79	0.00	0.01	0.00	0.38	0.01	0.00	0.00	0.04
Cuinar	0.00	0.97	0.06	0.00	0.03	0.00	0.00	0.00	0.00
Plats	0.00	0.02	0.75	0.01	0.04	0.00	0.00	0.00	0.00
Menjar	0.00	0.00	0.05	0.86	0.01	0.00	0.00	0.00	0.01
Altres	0.03	0.01	0.07	0.04	0.29	0.01	0.00	0.00	0.02
Social	0.00	0.01	0.00	0.01	0.00	0.98	0.00	0.00	0.00
Aspiradora	0.00	0.00	0.02	0.00	0.00	0.00	1.00	0.00	0.00
Televisior	0.00	0.00	0.02	0.03	0.00	0.00	0.00	0.99	0.00
Treballar	0.17	0.00	0.02	0.05	0.26	0.01	0.00	0.00	0.94

(b)

Taula B.2: Matrius de confusió del *fold* 1 del sistema Baseline. (a) Matriu no normalitzada. (b) Matriu normalitzada.

Valors de Predicció (%)

Valors Reals (%)	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	4332	0	2	0	73	1	0	0	300
Cuinar	0	1220	39	6	10	0	0	0	1
Plats	2	59	253	9	21	0	0	0	0
Menjar	5	1	44	457	12	1	0	0	52
Altres	93	12	29	11	303	1	0	0	59
Social	43	29	35	15	96	818	0	0	12
Aspiradora	0	0	0	0	0	0	264	0	0
Televisior	0	0	0	5	0	5	2	4680	0
Treballar	789	8	57	94	397	0	0	0	3235

(a)

Valors de Predicció (%)

Valors Reals (%)	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	0.82	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.08
Cuinar	0.00	0.92	0.09	0.01	0.01	0.00	0.00	0.00	0.00
Plats	0.00	0.04	0.55	0.01	0.02	0.00	0.00	0.00	0.00
Menjar	0.00	0.00	0.10	0.77	0.01	0.00	0.00	0.00	0.01
Altres	0.02	0.01	0.06	0.02	0.33	0.00	0.00	0.00	0.02
Social	0.01	0.02	0.08	0.03	0.10	0.99	0.00	0.00	0.00
Aspiradora	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00
Televisior	0.00	0.00	0.00	0.01	0.00	0.01	0.01	1.00	0.00
Treballar	0.15	0.01	0.12	0.16	0.43	0.00	0.00	0.00	0.88

(b)

Taula B.3: Matrius de confusió del *fold* 2 del sistema Baseline. (a) Matriu no normalitzada. (b) Matriu normalitzada.

Valors de Predicció

Valors Reals	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	4578	0	0	0	56	8	0	0	138
Cuinar	0	1222	61	0	37	0	0	0	0
Plats	0	4	282	23	41	4	0	0	6
Menjar	0	1	15	523	9	0	0	0	36
Altres	67	4	19	14	326	2	0	0	84
Social	0	0	4	0	0	1430	0	218	0
Aspiradora	0	0	0	0	0	0	228	0	0
Televisior	4	0	1	2	0	2	0	4831	0
Treballar	821	70	26	47	334	7	0	0	3447

(a)

Valors de Predicció (%)

Valors Reals (%)	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	0.84	0.00	0.00	0.00	0.07	0.01	0.00	0.00	0.04
Cuinar	0.00	0.94	0.15	0.00	0.05	0.00	0.00	0.00	0.00
Plats	0.00	0.00	0.69	0.04	0.05	0.00	0.00	0.00	0.00
Menjar	0.00	0.00	0.04	0.86	0.01	0.00	0.00	0.00	0.01
Altres	0.01	0.00	0.05	0.02	0.41	0.00	0.00	0.00	0.02
Social	0.00	0.00	0.01	0.00	0.00	0.98	0.00	0.04	0.00
Aspiradora	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Televisior	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00
Treballar	0.15	0.05	0.06	0.08	0.42	0.01	0.00	0.00	0.93

(b)

Taula B.4: Matrius de confusió del *fold* 3 del sistema Baseline. (a) Matriu no normalitzada. (b) Matriu normalitzada.

Valors de Predicció

Valors Reals	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	4499	0	1	1	27	3	0	0	109
Cuinar	0	1162	33	15	31	3	0	0	0
Plats	0	9	305	4	42	0	0	0	4
Menjar	4	0	5	525	16	0	0	0	30
Altres	144	9	7	1	272	1	0	0	82
Social	19	0	8	4	12	1142	0	3	20
Aspiradora	0	0	0	0	0	0	240	0	0
Televisior	4	0	0	0	0	4	0	4532	0
Treballar	683	0	1	18	154	8	0	0	3744

(a)

Valors de Predicció (%)

Valors Reals (%)	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	0.84	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.03
Cuinar	0.00	0.98	0.09	0.03	0.06	0.00	0.00	0.00	0.00
Plats	0.00	0.01	0.85	0.01	0.08	0.00	0.00	0.00	0.00
Menjar	0.00	0.00	0.01	0.92	0.03	0.00	0.00	0.00	0.01
Altres	0.03	0.01	0.02	0.00	0.49	0.00	0.00	0.00	0.02
Social	0.00	0.00	0.02	0.01	0.02	0.98	0.00	0.00	0.01
Aspiradora	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Televisior	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Treballar	0.13	0.00	0.00	0.03	0.28	0.01	0.00	0.00	0.94

(b)

Taula B.5: Matrius de confusió del *fold* 4 del sistema Baseline. (a) Matriu no normalitzada. (b) Matriu normalitzada.

Annex C

Resultats de les modificacions

Resultat del conjunt de test en l'enfinestrament de 0.500 s

Scene	Fold 1 (Baseline)	Fold 1 (500 ms)
Absència	84.23	75.36
Cuinar	96.52	96.03
Rentar plats	78.56	77.14
Menjar	85.54	79.10
Altres	35.43	32.96
Activitat social	96.48	95.35
Neteja amb aspiradora	97.66	99.79
Mirar televisior	99.40	99.16
Treballar	82.96	71.14
Mitjana	**84.09**	**80.67**

Taula C.1: Comparació de resultats del conjunt de test, del *fold* 1, del sistema Baseline, respecte a la modificació d'enfinestrament de 500 ms, per a cada una de les classes a avaluar.

Scene	Fold 1 (Baseline)	Fold 1 (80 ms)
Absència	84.23	82.37
Cuinar	96.52	96.85
Rentar plats	78.56	78.83
Menjar	85.54	82.89
Altres	35.43	36.48
Activitat social	96.48	96.57
Neteja amb aspiradora	97.66	99.17
Mirar televisior	99.40	99.47
Treballar	82.96	81.65
Mitjana	**84.09**	**83.81**

Taula C.2: Comparació de resultats del conjunt de test, del *fold* 1, del sistema Baseline, respecte a la modificació d'enfinestrament de 80 ms, per a cada una de les classes a avaluar.

Millora del Sistema Baseline *Fold* 1 (500 ms):

Valors Reals \ Valors de Predicció	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	3842	1	13	2	357	4	0	0	513
Cuinar	0	1211	43	0	29	0	0	0	1
Plats	0	14	302	3	31	2	0	0	4
Menjar	20	0	35	424	47	0	0	0	46
Altres	149	10	22	12	250	2	0	0	75
Social	10	1	4	16	3	975	0	26	1
Aspiradora	0	0	0	0	0	0	240	0	0
Televisior	4	1	4	18	3	19	1	4525	1
Treballar	1440	0	4	25	277	7	0	0	2951

(a)

Valors Reals (%) \ Valors de Predicció (%)	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	0.70	0.00	0.03	0.00	0.36	0.00	0.00	0.00	0.14
Cuinar	0.00	0.98	0.10	0.00	0.03	0.00	0.00	0.00	0.00
Plats	0.00	0.01	0.71	0.01	0.03	0.00	0.00	0.00	0.00
Menjar	0.00	0.00	0.08	0.85	0.05	0.00	0.00	0.00	0.01
Altres	0.03	0.01	0.05	0.02	0.25	0.00	0.00	0.00	0.02
Social	0.00	0.00	0.01	0.03	0.00	0.97	0.00	0.01	0.00
Aspiradora	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Televisior	0.00	0.00	0.01	0.04	0.00	0.02	0.00	0.99	0.00
Treballar	0.26	0.00	0.01	0.05	0.28	0.01	0.00	0.00	0.82

(b)

Taula C.3: Matrius de confusió del *fold* 1 del sistema modificat amb enfinestrament de 500 ms. (a) Matriu no normalitzada. (b) Matriu normalitzada.

Millora del Sistema Baseline *Fold* 1 (80 ms):

Valors Reals \ Valors de Predicció	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	4079	6	37	3	337	5	0	0	265
Cuinar	0	1231	27	0	25	0	0	0	1
Plats	0	9	311	0	32	0	0	0	4
Menjar	15	0	27	448	29	0	0	0	53
Altres	137	5	23	14	263	4	0	0	74
Social	13	3	0	13	4	986	0	17	0
Aspiradora	0	1	0	0	0	0	239	0	0
Televisior	4	0	4	12	3	5	3	4545	0
Treballar	924	3	4	19	229	6	0	0	3519

(a)

Valors Reals (%) \ Valors de Predicció (%)	Abs.	Cuin.	Plats	Men.	Alt.	Soc.	Asp.	TV	Treb.
Absència	0.79	0.01	0.09	0.01	0.37	0.01	0.00	0.00	0.07
Cuinar	0.00	0.98	0.06	0.00	0.03	0.00	0.00	0.00	0.00
Plats	0.00	0.01	0.72	0.00	0.04	0.00	0.00	0.00	0.00
Menjar	0.00	0.00	0.06	0.88	0.03	0.00	0.00	0.00	0.01
Altres	0.03	0.00	0.05	0.03	0.28	0.00	0.00	0.00	0.02
Social	0.00	0.00	0.00	0.03	0.00	0.98	0.00	0.00	0.00
Aspiradora	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00
Televisior	0.00	0.00	0.01	0.02	0.00	0.01	0.01	1.00	0.00
Treballar	0.18	0.00	0.01	0.04	0.25	0.01	0.00	0.00	0.90

(b)

Taula C.4: Matrius de confusió del *fold* 1 del sistema modificat amb enfinestrament de 80 ms. (a) Matriu no normalitzada. (b) Matriu normalitzada.